

15 May 1992

Communications-Computer Systems

COMPUTER SOFTWARE STANDARDS

This regulation prescribes policy and procedures for software standards within Headquarters Air Force Global Weather Central (HQ AFGWC). It applies to all HQ AFGWC personnel involved in the development and maintenance of computer software.

1. General. Software standards are tools used to achieve quality software. Quality software is defined as that which performs the required function, satisfies the user, and has acceptably low life-cycle costs. Software standards improve productivity, testability, reliability, maintainability, and portability of software throughout the software life cycle. Software standards are needed to ensure the various software components (designed, coded, and documented by different individuals) achieve compatible and consistent overall system design, and uniformity. Standards provide the project team with a means of control.

2. HQ AFGWC Software Standards. The HQ AFGWC Software Standards codify the standards adopted by HQ AFGWC. All standards are reviewed by the Software Standards Working Group (SSWG) as authorized by the Configuration Control Board (CCB).

a. The "Standards" include a table of contents and the following groups of standards:

- (1) General (Atch 1).
- (2) Analysis/Design (Atch 2).
- (3) Documentation (Atch 3).
- (4) Configuration Management (Atch 4).
- (5) Programming (Atch 5).
- (6) Review/Audit (Atch 6).
- (7) Operational Task (Atch 7).

b. The format of individual standards is as follows:

- (1) Applicability.
- (3) References (optional).
- (3) Requirements.
- (4) Notes (optional).

3. **Applicability.** The Standards apply to all software developed at HQ AFGWC, including contract-developed software. These standards apply regardless of whether the software is targeted for a mainframe, mini-computer, or microcomputer workstations. This does not include software purchased for personal computers (PCs) (i.e., Enable, Chart, etc.). Software submitted for the Quality Assurance Audit (QAA) must conform to the standards that are in effect at

the time of the QAA. Waivers are granted on a case-by-case basis pending the severity of the impacts. General Standard #1 gives a detailed description of applicability. The individual standards further refine their applicability.

4. Procedures.

a. The Automation Request and Standardization Office (HQ AFGWC/SYR), will maintain the standards for HQ AFGWC.

b. HQ AFGWC Branch/Staff Agency Chiefs will ensure all personnel responsible for the development or maintenance of software (programmers, project managers, and others) become familiar with the current/proposed standards for their impact on projects. HQ AFGWC Branch Staff Agency Chiefs will ensure compliance with the standards. Programming functional and the Systems Management Branch staff (AFGWC/SYP) work areas should conduct periodic reviews of their software development and maintenance efforts to ensure all standards are being met. DO NOT wait until the QAA to find out whether or not the code meets standards.

c. Anyone desiring to initiate changes, additions, or deletions to the standards will forward his or her proposal (Atch 8), in writing, to the HQ AFGWC SSWG Recorder for distribution to working group representatives. The request should include the proposed change, why the change is needed, and what the benefits will be if the change is implemented. The SSWG will review the proposal and if necessary, implement changes. Individuals submitting proposed changes will be the point of contact (POC) for that proposal. The POC will be asked to attend the SSWG meeting when his/her proposal is scheduled for review. Following SSWG approval, HQ AFGWC/SYR will forward the new or revised standard to HQ AFGWC Information Management for inclusion in the next publication of AFGWCR 700-2.

d. A standard may be waived for a project by submitting an AFGWC Form 12, Software Standards Waiver Request (Atch 9). Instructions for completing the waiver request are found on the back of the form. Waiver requests will be sent through the requesting branch/staff agency chief and Pro-

supersedes AFGWCR 700-2, 1 April 1991 (see signature page for summary of changes).

No. Printed Pages: 98

OPR: SYR

Approved By: Colonel R. E. Townsend

Editor: P. L. Sheaffer

Distribution: F;X

AWS/XT, Scott 1

Program Managers (PM) to HQ AFGWC/SYR. Waiver requests for contractor software will be reviewed by Chief, Systems Division (SY) prior to release. Every effort will be made to submit the waiver request prior to the QAA. Standards violations identified during the QAA may be addressed by submitting a waiver request. All violations will be corrected with the next revision of the software. Software Branch (HQ AFGWC/SYS/Quality Assurance), or their designated representative, will verify that all violations noted during the QAA have been waived. If any violation was not waived, then all violations will be corrected and a new QAA will be scheduled and conducted. HQ AFGWC/SYR will approve or disapprove each waiver request. If the branch/staff agency chief does not concur with HQ AFGWC/SYR decision, he/she may request the CCB review the waiver request and make a ruling.

e. HQ AFGWC/SYR will forward a monthly listing to HQ AFGWC/SY, Chief, Systems Management Branch

(SYP), and Chief, Software Branch (SYS) containing all open waivers for standards violations. This listing is used to ensure all violations are corrected with the next revision of the software. HQ AFGWC/SYS/Quality Assurance will notify HQ AFGWC/SYR when previously waived violations have been corrected.

f. To close an open waiver after action has been completed, sign a copy of the waiver in the date action closed box and send a copy to HQ AFGWC/SYS/Quality Assurance and HQ AFGWC/SYR.

5. Related Publications.

a. DOD 7935.1-STD, Automated Data Systems (ADS) Documentation Standards.

b. AFR 700, Communications-Computer, series regulations.

c. HQ AFGWC 171, Automatic Data Processing Systems and Procedures, series regulations.

OFFICIAL

THOMAS E. SIELAND, Colonel, USAF
Commander

WILLIAM S. KELLY: CMSgt, USAF
Chief, Information Management

9. Atch

1. General Standards
2. Analysis/Design Standards
3. Documentation Standards
3. Configuration Management Standards
5. Program Standards
6. Review/Audit Standards
7. Operational Task Standards
8. Sample Standards Change Proposal
9. Sample Waiver Request Form

SUMMARY OF CHANGES

Rewrite of standards G-SI-A1, G-S2, A-SI, A-S2, D-S1, D-S2, D-S3, D-S5, P-S3, P-S4, P-S14, P-S15, P-S17, P-S18, R-S6, and R-S10.

GENERAL STANDARDS

G-S1 - Applicability of Standards	1 Apr 91
Specifies when software will comply with standards.	
G-S2 - Definitions	1 Apr 91

General Standards

G-S1, APPLICABILITY OF STANDARDS

1. **Applicability.** AFGWC software standards apply to all new and changed software, except where exceptions or extensions are specified in a particular standard.
2. **Reference.** N/A.
3. **Requirement:**
 - a. **New software.** The standards apply to all new programs, subroutines, and runstreams.
 - b. **Changed software.** The extent to which existing software must be brought into compliance with the standards during modification depends on the extent of the change and on the nature of the standard. See the Retrofit Table on the following page (G-S1-A1).
 - (1) The percentages refer to the number of lines of executable code which are involved in a single change. For a list of executable and nonexecutable FORTRAN statements see the applicable FORTRAN PRM. Cosmetic changes (those that do not affect the program's function) do not count as changes to executable code. Cosmetic changes include, but are not limited to the following:
 - (a) Adding/deleting spaces.
 - (b) Adding/deleting blank lines.
 - (c) Adding/deleting comments.
 - (d) Changing indentation.
 - (e) Changing labels, as long as all references (GO TOs, JUMPs, etc) are changed to refer to the new labels.
 - (2) The retrofit table applies to the percent of change in each subsystem, program, subroutine, and module. Therefore, a complete overhaul of a module may automatically imply a significant revision of the program even if all the changes are in the one module.
 - (3) Where retrofit of standards is not performed, the programmer will:
 - (a) make all new or modified lines conform to standards, or
 - (b) make the modified code consistent with the existing code, thereby enhancing readability.
 - (4) The retrofit table reflects the minimum acceptable compliance. Programmers are encouraged to increase the compliance of software with the standards whenever they are able to do so.
 - c. **Exception:** Unique situations sometimes arise whereby compliance with a standard may not be warranted. In such a case, the section chief may request a waiver from SYR.
 - d. **Non-FORTRAN.** Some standards are meaningful only for software written in FORTRAN. They are so marked in the retrofit table.

G-SI-AI, RETROFIT TABLE					
<u>STANDARD</u>		<u>FORTRAN</u> <u>ONLY</u>	<u>ANY</u> <u>(GT 0%)</u>	<u>SIG</u> <u>REVIS</u> <u>(GT 20%)</u>	<u>COMPL</u> <u>OVERHAUL</u> <u>(GE 40%)</u>
A-S1	Isolation of Function	YES			X
A-S2	Top-Down Structured Design				X
D-S1	Process Narrative		X	X	X
D-S2	Prologue for Routines		X	X	X
D-S3	Comments			X	X
D-S4	Minimum Documentation Requirements				X
D-S5	Software Listings		X	X	X
D-S6	INFORM Messages		X	X	X
D-S7	Operator's Reference Guide (ORG)		X	X	X
M-S1	Elements Added or Executed				X
M-S2	Subsystem and Production Program File Naming Standard				X
P-S1	Indentation	YES		X	X
P-S2	Console Message/Response				X
P-S3	Variable and Constant Names and Declarations	YES			X
P-S4	Statement Numbering	YES			X
P-S5	ASSIGN Statement	YES			
P-S6	Arithmetic IF Statement	YES			X
P-S7	Error Checking			X	
P-S8	GO TO Statements	YES			X
P-S9	Library Processes				X
P-S10	Use of Parentheses	YES		X	X
P-S11	FORTRAN Extensions	YES		X	
P-S12	Use of @JUMP Command in Runstreams and ADD Elements				X
P-S13	Length of Arithmetic Assignment Statements	YES			
P-S14	Internal Organization of Routines			X	X
P-S15	Routine Size	YES			X
P-S16	Assigning Tape Files		X	X	X
P-S17	Use of INCLUDE Statement and the FORTRAN PROC				X
P-S18	COMMON Blocks				X

General Standards

GS2, DEFINITIONS

1. **Applicability:** This standard gives the accepted definitions of the following terms and acronyms within the Software Standards.
2. **Reference.** N/A.
3. **Requirement.** The following definitions and acronyms are provided for standard usage. *Italicized words within a definition have their own definition.*

Absolute: A self-contained, executable software entity consisting of one or more routines, also referred to as a program

Add Element: A List of records containing executive control statements, but no @RUN card.

AFGWC*ASCIREL: A file containing the FORTRAN 77 relocatables for use with FORTRAN 77 programs. These relocatables are database and local utility routines.

AFGWC*ASCIREL-65K: A file similar to ASCIREL. Contains relocatables that are compatible with programs larger than 65K

AFGWC*USERSREL: A file containing FORTRAN V and MASM relocatables for use with FORTRAN V programs. These relocatables are database and local utility routines.

ANSI: American National Standards Institute

Argument: Any value of an independent variable.

ASCII FORTRAN: UNISYS version of ANSI Standard FORTRAN 77.

CDR: Critical Design Review. See R-S5

CCA: Configuration Control Audit. See R-S11.

CPC: Computer Program Component. A functionally or logically distinct part of a CPCI. The structure of CPC identifiers is:

For absolutes:

FF/SSS/NNNNNN

FF - Family level identifier.

SSS - Subfamily level identifier.

NNNNNN - Unique absolute identifier (such as its name but limited to a maximum of six characters).

For Utilities: (two forms)

UT/SSS/NNNNNN

UT - Utility family.

SSS - ASC or USE

NNNNNN - Utility identifier

FF/UTI/NNNNNN

FF - Family level identifier

UTI - Utility Subfamily

NNNNNN - Utility identifier

For products:

AP/CCC/MMM/NNNNNN

MP/CCC/MMM/NNNNNN

AP - Automated Product family.

MP - Manual Product family.

CCC - Primary customer.

MMM - Method of delivery.

NNNNNN - Product identifier (such as its name but limited to a maximum of six characters).

For other components:

FF/SSS/TTT/NNNNNN

FF - Family level identifier.

SSS - Subfamily level identifier.

TTT - Type identifier:

AAA - Add element.

DDD - Data element.

DFI - Data file.

RNS - RuNStream.

RTN - RouTine.

UTI - UTILity.

NNNNN - Unique identifier (such as it's name but limited to a maximum of six characters)

CPCI: Computer Program Configuration Item An automated data system (ADS) or portion of an ADS that is designated for configuration management. The structure for CPC identifiers is:

FF/ SSS where:

FF - Family level identifier.

SSS - Subfamily level identifier.

Database View: the logical (and/or physical) configuration of data items as passed between one or more software units (that unit's "view" of the database).

Data Element: A list of records containing no ECL statements.

DS: Database Specification. See D-S4.

EOT: Extended Operational Test. An on-Line test lasting 24 hours or more.

Executable Process: an executable logical software unit. (An absolute on a UNISYS. Consists of a main routine in FORTRAN, a routine with a PROGRAM statement), and all subroutines and subprocesses, if any.

Family: The highest level in the hierarchy of CPCIs. Normally, a grouping of related subfamilies and parallel to the programming organization hierarchy.

FD: Functional Description. See D-S4.

File: A organized collection of information. There are two kinds of files; program and data.

FLASH/AFLASH: Documentation for library routines in AFGWC*USERSREL. and AFGWC*ASCIREL(765K)

FORTRAN V: UNISYS version of ANSI Standard FORTRAN 66.

Function: can refer to functionality at any level: "add two registers together" (an assembler statement); a very small, very specific function; while "produce a global spectral model forecast" (a subsystem) is very large, very general function.

FCA: Functional Configuration Audit. See R-S8.

Library Process: a logical software unit in a library. Consists of a main routine, and all subroutines and subprocesses, if any. Normally implements one small function.

Main Routine: The starting point (often the Entry Point) for the process. The main routine will almost always have the same name as the process. Examples are a runstream or the FORTRAN routine with a PROGRAM statement.

MM: Maintenance Manual. See D-S4.

Object: Any set of information which can be treated as a basic unit by the operating system. For example, a data file, the code for a program, a runstream, etc.

ORG: Operator's Reference Guide. See D-S7.

OSCL: Operating System Control Language. UNISYS ECL, IBM JCL, DEC VAX DCL, and batch files on an IBM/PC are examples. (Also see runstream)

OSCL Process: a logical OSCL software unit. Consists of a main routine, and all subroutines (usually add elements on a UNISYS) and subprocesses (usually absolutes on a UNISYS), if any. A "startable OSCL process" contains a runstream as the main routine.

PCA: Physical Configuration Audit. See R-S9.

PDR: Preliminary Design Review. See R-S4.

PROC: A section of code that is included in the symbolics at completion time. Usually redundant code (e.g., common blocks).

Process: Consists of a set of objects required to perform a particular task and information about the current state of the task. Normally implements one function. A Large process with many routines and subprocesses normally implements a Large, general function.

PS: Program Specification. See D-S4.

PT: Test Plan. See D-S4.

PVR: Product Verification Review. See R-S6.

QAA: Quality Assurance Audit. See R-S10.

RD: Data Requirements Document. See D-S4.

Routine: a physically separate software unit.

a. A routine is:

(1) A separately compilable software unit, such as a FORTRAN SUBROUTINE, a main FORTRAN routine (PROGRAM), a FORTRAN FUNCTION, a FORTRAN PROC, or a MASM element.

(2) Or, a separate element of OSCL, Like an ADD elt, or a runstream

b. A routine does not include any other software units which are used by that routine.

c. There is a special type of routine called a main routine.

RT: Test analysis Report. See D-S4.

Runstream: A list of records containing ECL statements and containing an @RUN statement as the first record.

SDR: System Design Review. See R-S3.

SRR: System Requirements Review See R-S2.

SS: System/Subsystem Specification. See D-S4.

Subfamily: The first subgrouping within a family, normally a group of related absolutes.

Subprocess: a process used by another process or routine.

Subroutine: a routine used by another routine.

SVR: System Validation Review See R-S7.

UM: User's Manual. See D-S4.

Utility: A routine used by multiple families, subfamilies, or absolutes.

4. Note. Definitions for routine, subroutine, main routine, and process are arbitrary. Don't attempt to apply preconceived meanings to these definitions.

ANALYSIS/DESIGN STANDARDS

A-S1 - Isolation of Function

1 Apr 91

Defines and describes isolation of function to enhance software portability.

A-S2 - Top-Down, Structured Design

1 Apr 91

Specifies required software design characteristics.

Analysis/Design Standards

A-S1, ISOLATION OF FUNCTION

1. **Applicability.** All new or extensively-modified software.
2. **Reference.** N/A.
3. **Requirement.** To the maximum extent possible, software will reflect isolation of Function, particularly in those elements (I/O, data manipulation, job control, and environment definition) that are most difficult to transport.
 - a. All binary bit packing and unpacking will be performed in routines that are separate from application routines.
 - b. File structures must be easily transferable to a 32-bit computer word architecture.
 - (1) Binary Packed data stored in a UNISYS 36-bit word must also fit in a 32-bit word
 - (2) Packing strategies will be byte oriented. Quarter-word-packed data must fit into 8 bits. Half-word packed data must fit into 16 bits.
4. **Notes:**
 - a. Isolation of function is related to the practice of increased program modularity in terms of function and logic. The difference is one of emphasis: particular attention is to be directed to those features of the design which reflect underlying characteristics of the hardware (e.g., 36-bit word architecture) or of data (e.g., current AFGWC database formats). Segregating these features reduces the impact of change in the processing environment.
 - b. Isolating the functions of packing and unpacking data means that most routines can work with data in whole words. This makes software less dependent on the underlying data structure and the machine architecture.

Analysis/Design Standards

A-S2, TOP-DOWN, STRUCTURED DESIGN

1. **Applicability.** All new software, and changed software per Standard G-S1
2. **References:**
 - a. Meilir Page-Jones, The Practical Guide to Structured Systems Design, 1980, Yourdon Press.
 - b. Brian Dickson, Developing Structured Systems, 1980 Yourdon Press.
 - c. Yourdon and Constantine, Structured Design, 1975, Yourdon Press.
3. **Requirement:** Software shall be developed in a top-down structured manner.
 - a. Top-down design starts with the problem/requirement and breaks it down into successively smaller parts. Design proceeds from the top (most general) to the bottom (most detailed). The resulting design mimics/reflects the nature of the problem and is readable from the top down.
 - b. Structured software is composed of separate functional processes. Each process may invoke subordinate subprograms in order to perform a single function. Subprograms are arranged in a hierarchy, with control flowing from the top to the bottom
 - c. Well-designed, top-down, structured software has the characteristics Listed below. AFGWC software will have these characteristics to the greatest extent practicable:
 - (1) High strength (cohesion). A process performs a single function or small related functions that require the same data and pass data from one step to another.
 - (2) Loose connectivity (coupling). Connections between processes or routines are minimized, and they are obvious! Passing data via arguments is preferred; global variables and common areas are less desirable.
 - (3) Limited data exchange. A minimum amount of data is passed between processes or routines.
 - (4) Single entry and exit points. See P-S14, Internal Organization of Modules.
 - (5) Limited span of control. Routines directly control a maximum of seven subordinates.
 - (6) Span of control encompassing the scope of effect. Modules whose execution depends on decisions made in another module are subordinate to it.
 - (7) Small size. See P-S15, Routine Size.
 - (8) No state memory. Execution does not depend on anything that happened on a previous invocation; the same inputs always produce the same outputs.
4. **Notes:**
 - a. Top-down, structured software is achieved through structured techniques and tools such as data flow diagrams, process descriptions, data dictionaries, and structure charts.
 - b. Top-down, structured design has these advantages:
 - (1) It allows top-down coding, testing, and implementation;
 - (2) Allows early detection and cheaper correction of design problems;
 - (3) When developing a given process, allows consideration of other processes as black boxes;
 - (4) Increases ease of debugging and maintainability; and
 - (5) Results in fewer original errors.

DOCUMENTATION STANDARDS

D-S1 - Process Narrative	1 Apr 91
Defines minimum identification to be embedded in each process developed or modified extensively at AFGWC.	
D-S2 - Prologue for Routines	1 Apr 91
Defines minimum identification to be embedded in each routine developed or modified extensively at AFGWC.	
D-S3 - Comments	1 Jul 89
Defines uniform format for using comment statements in FORTRAN software.	
D-S4 - Minimum Documentation Requirements	20 Apr 90
Specifies minimum documentation requirements as a function of project cost.	
D-S5 - Software listings	1 Apr 91
Defines what goes into software listings.	
D-S6 - INFORM Message	1 Jul 89
Defines the format and content of INFORM messages.	
D-S7 - Operator's Reference Guide (ORG)	1 Jul 89
Requires an ORG on all production runstreams and TIP transactions.	

Documentation Standards

D-ST, PROCESS NARRATIVE

1. Applicability:

- a. In accordance with Standard G-SI.
- b. When applied to modified software, mandatory retrofit is limited. A process narrative must be created or updated to the extent possible with the knowledge the programmer has gained from working with the code. No special research into the code is required to retrofit a process narrative.
- c. For processes without narratives, one must be created. For processes with existing narratives or prologues in another format, the existing format may remain if all required information is present. Reformatting an existing narrative is required for processes undergoing complete overhaul, as defined in Standard G-SI.

2. Reference. AFGWC 700-2 Atch 4 (D-S2), Atch 5 (P-S14)

3. Requirement.

- a. Each process must have a narrative containing the entries in the order listed below (Additional entries may be added):

- (1) Name - in plain English. Will usually be the same as the name of the main routine.
- (2) CPC - provide the CPC name, if applicable. (If not applicable, delete this entry.)
- (3) Purpose - brief description of reason for the process's existence.
- (4) Method - general description of how the process accomplishes its purpose. Describe any execute/SETC options and their effects. Delete this entry for processes which contain no executable code.
- (5) Reference(s) - significant reference materials useful in understanding the process. If none, so state.
- (6) Terms and Abbreviations - define all terms or abbreviations not commonly used or not in G-S2. If none, so state.
- (7) Common Blocks - list and describe all common blocks that appear in the process. If none, so state.
- (8) Hierarchy Chart - provide a hierarchy chart of all routines and subprocesses used. If there are no subroutines or subprocesses, delete this entry. This can reference an external chart (e.g. a CASE tool generated hierarchy chart) in lieu of including it here.
- (9) Object - list and describe all objects that make up the process.
 - (a) OSCL elements - list and describe all OSCL elements associated with this process; i.e., runstreams and add elements
 - (b) Routines - alphabetically list and describe.
 - (c) Files Accessed - fully identify all files and tray data sets used by the process. Give both the FORTRAN logical unit number and System File name/data set designations if known and applicable. Use R to indicate R FROM FILE and W to indicate WRITE TO FILE. Give a brief description of each file, and a reference to detailed information. If none, so state.
 - (d) Other - list and describe any other objects associated with this process.
- (10) Updates to Routines - list at least the date of the update and the name of the routine. This is applicable to all routines in the process. It is not applicable to separate subprocesses used by this process.

- b. The narrative will be maintained in a separate file, such as a word processing file, or a CASE tool file.

4. Notes: A generic shell is available from HQ AFGWC/SYS/Quality Assurance or HQ AFGWC/SYR.

Documentation Standards

D-S2, PROLOGUE FOR ROUTINES

1. Applicability:

- a. In accordance with Standard G-S1.
- b. When applied to modified software, mandatory retrofit is limited. A prologue must be added or updated to the extent possible with the knowledge the programmer has gained from working with the code. No special research into the code is required to retrofit a prologue.
- c. For routines without prologues, a prologue must be added. For routines with existing prologue in another format, the existing format can remain if all required information is present. Reformatting an existing prologue will be required only for routines undergoing complete overhaul, as defined in Standard G-S1.
- d. Some programs use a program version message that is updated whenever a subprogram is altered. This standard is not applicable to changes that consist only of an update to the program version message.
- e. Data files and data elements are excluded.
- f. This standard is not applicable if the only change to the routine is an update to a process narrative that is at the beginning of the routine.

2. References: AFGWC 700-2 Atch 3 (D-S1), Atch 4 (M-S2), and Atch 5 (P-S14)

3. Requirements: Prologue entries will be grouped together, and will be set off from the rest of the Listing by a border consisting of a double line of non-blank characters (see D-S14). Short key-in runstreams with four or less lines may be identified with a prologue containing only the runstreams CPC (see M-S2). All others will consist of the following entries except as noted (additional entries may be added):

- a. Name - What the routine name means spelled out in plain English (not the six character FORTRAN name). Or, Data File Specification view name for a logical database record.
- b. Purpose - Brief description of reason for the routine's existence.
- c. Method - General description of how the routine accomplishes its purpose. A routine with no executable code may delete this entry.
- d. Process Narrative - The name of the process narrative for this routine and its location.
- e. Reference(s) - Significant reference materials useful in understanding the routine. If none, so state.
- f. Interface - Define the interface with the calling routines. Any listed variable which is in a common block must include the name of the common block. A routine with no executable code may delete these entries. OSCL routines with no execute options may delete these entries. Entries will be made in the following manner:
 - (1) Input variables - List all input variables. Routines with no input variables use "NONE".
 - (2) Output variables - List all output variables. Routines with no output variables use "NONE".
 - (3) Files Accessed - fully identify all files and Cray data sets used by the object. Give both the FORTRAN logical unit number and system file name/date set designations if known and applicable. Use R to indicate READ FROM FILE and W to indicate WRITE TO FILE. Give a brief description of each file, and a reference to detailed information. If none, so state.
- g. Updates - First entry is the date the routine was written, and the programmer's name and functional work areas.. Subsequent entries list the date and describe the changes made. In addition, they will include the programmer's name and HQ AFGWC/SYS/Quality Assurance or HQ AFGWC/SYR.

4. Notes: A generic shell is available from HP AFGWC/SYS/Quality Assurance or HQ AFGWC/SYR.

Documentation Standards

D-S3, COMMENTS

1. **Applicability.** In accordance with Standard G-S1.
2. **Reference.** Software Quality Assurance, Thomas McCabe Associates. McCabe Press, 1980.
3. **Requirement.** Comments must be set off from code in a uniform manner. They must not detract from understanding the flow of the code.
 - a. Place block comments at the beginning of each block of code. The block comments describe the function of the code block.
 - b. Set off block comments with a border. The format consists of one or more lines of blanks, one or more lines of distinguishing characters, the comment itself, another line of distinguishing characters, followed by one or more line of blanks.
 - c. All commented out code, except clearly documented diagnostic code, will be removed.
 - d. In-line comments:
 - (1) For Assembler code, there is no restriction on in-line comments.
 - (2) For FORTRAN code, in-line comments will only be used in FORTRAN variable and constant declaration sections.
4. **Notes:**
 - a. D-S3-A1 is an example of an acceptable format (other formats may be acceptable).
 - b. Indenting the text of the comment to the same extent as the code is recommended. The comment begins with the first non-blank character after the comment indicator.

D-S3-A1,EXAMPLE OF AN ACCEPTABLE COMMENT FORMAT

***** BEGINNING OF MODULE *****

```

C -----
C INTERPOLATE SIGMA LAYER POTENTIAL TEMPERATURE AND SPECIFIC
C HUMIDITY TO THE RAOB STATION
C -----

```

DO 20 STN = 1, LSTSTN

```

C -----
C SET THE I-AXIS AND J-AXIS INDICES AND DISTANCE DELTAS
C -----

```

```

I = INT( STNI(STN) )
DELI = STNI(STN) - REAL(I)
IP1 = I + 1

```

```

J = INT( STNJ(STN) )
DELJ = STNJ(STN) - REAL(J)
JP1 = J + 1

```

```

C -----
C LOOP THRU EACH SIGMA LEVEL AT THIS STATION
C -----

```

DO 10 K = 1, KLEV

```

C -----
C HORIZONTALLY INTERPOLATE THE POTENTIAL TEMPERATURE
C -----

```

```

LJ = TS(I,J,K) + DELI*( TS(IP1,J,K) - TS(I,J,K) )
LJP1 = TS(I,JP1,K) + DELI*( TS(IP1,JP1,K) - TS(I,JP1,K) )
PTSIG(STN,K) = LJ + DELJ*( LJP1 - LJ )

```

```

C -----
C HORIZONTALLY INTERPOLATE THE SPECIFIC HUMIDITY
C -----

```

```

LJ = QS(I,J,K) + DELI*( QS(IP1,J,K) - QS(I,J,K) )
LJP1 = QS(I,JP1,K) + DELI*( QS(IP1,JP1,K) - QS(I,JP1,K) )
SHSTN(STN,K) = LJ + DELJ*( LJP1 - LJ )

```

10 CONTINUE

20 CONTINUE

```

C -----
C DERIVE THE FORECAST RAOB'S RH*, TEMPERATURE, VIRTUAL TEMPERATURE
C -----

```

DO 40 STN = 1, LSTSTN

DO 30 K = 1, KLEV

```

C -----
C CONVERT SPECIFIC HUMIDITY TO RH*
C -----

```

```

VAPOR = EW( PTSIG(STN,K) )
QSAT = WS( VAPOR , PRSMY(K)*PSTN(STN) )
RHSTN(STN,K) = SQRT(1.0 - SHSTN(STN,K)/QSAT)

```

```

C -----
C CONVERT POTENTIAL TEMPERATURE TO TEMPERATURE
C -----

```

```

TSIG(STN,K) = PTSIG(STN,K)*PIMY(K)*(PSTN(STN)*0.001)**0.286
VTSIG(STN,K) = TSIG(STN,K) * (1.0 + 0.61 * SHSTN(STN,K))

```

30 CONTINUE

40 CONTINUE

***** END OF MODULE *****

D-S3-A2, EXAMPLES OF ACCEPTABLE COMMENTED-OUT CODE

1. Diagnostic Code:

```

C*****
C*
C*   DIAGNOSTIC CODE TO DUMP THE ARRAYS BEING PROCESSED.
C*   IF ITYPE = 1, PRINT AARRAY, OTHERWISE PRINT BARRAY.
C*
C*****
C*
C*   PRINT *, ' SUBROUTINE XXX --DIAGNOSTIC PRINT'
C*   PRINT *, ' ITYPE IS ', ITYPE
C*
C*   IF (ITYPE.EQ.1) THEN
C*       PRINT *, ' AARRAY -- '
C*       PRINT *, ' SCFOBS, UAOPS, ACFOBS, ROCOPS, SATOPS, XOPS'
C*       DO 30 I = 1,10
C*           DO 20 J = 1,10
C*               DO 10 K = 1,10
C*                   WRITE (6, 40) AARRAY(I,J,K,1), AARRAY(I,J,K,7),
C*   &                          AARRAY(I,J,K,5), AARRAY(I,J,K,12),
C*   &                          AARRAY(I,J,K,15), AARRAY(I,J,K,22)
C* 10          CONTINUE
C* 20          CONTINUE
C* 30          CONTINUE
C*
C*   ELSE
C*       PRINT *, ' BARRAY -- '
C*       PRINT *, ' MCOPS, KEYOPS, MDUOPS, SEOPS, CLUOPS, BOBS'
C*       DO 80 I = 1,10
C*           DO 70 J = 1,10
C*               DO 60 K = 1,10
C*                   WRITE (6, 50) AARRAY(I,J,K,1), AARRAY(I,J,K,7),
C*   &                          AARRAY(I,J,K,5), AARRAY(I,J,K,12),
C*   &                          AARRAY(I,J,K,15), AARRAY(I,J,K,22)
C* 60          CONTINUE
C* 70          CONTINUE
C* 80          CONTINUE
C*   ENDIF
C*
C*****
C*
C*   END OF DIAGNOSTIC CODE
C*
C*****
90 STOP

```

2. DEBUG Statement:

```

C*****
C*
C*   DEBUG INIT,TRACE,SUBTRACE,UNIT(13)
C*
C*****
END

```

Documentation Standards

D-S4, MINIMUM DOCUMENTATION REQUIREMENTS

1. **Applicability.** All software development, modification, and maintenance projects.
2. **References:**
 - a. DOD 7935.1-STD, Automated Data Systems (ADS) Documentation Standards
 - b. AFR 700-4, Vol I.
 - c. AFGWCR 700-1.
 - d. AFGWCR 171-5.
3. **Requirements:**
 - a. Document software at a detail level compatible with the overall size and complexity of the project. D-S4-A1 summarizes these requirements with a Decision Logic Table (DLT) and indicates a reference for each document.
 - b. Develop a project plan, to include documentation to be developed or updated, early in the project. The plan is approved during the System Requirements Review (SRR). It may require the documents listed here for smaller projects, and/or require additional products such as data flow diagrams and hierarchy charts.
4. **Note.** Incorporation of graphic products such as data flow diagrams into documentation is highly recommended.

D-S4-A1, REQUIRED DOCUMENTATION BASED ON ESTIMATED PROJECT COST

DOCUMENTATION REQUIRED:	PROJECT COST			
	CI-500wh	501-1728wh	1729wh-17280wh	More than 17280wh
Users Manual (UM) (Note 1)	2	1	1	1
Operator's Reference Guide (ORG)	3	3	3	3
Program Maintenance Manual (MM)	Note 2	1	1	1
Test Plan (PT)	2	1	1	1
Test Analysis Report (TAR)	2	1	1	1
Functional Description (FD)			1	1
System/Subsystem Spec (SS)			1	1
Data Requirements Document (RD)				1
Program Specification (PS)				1
Database Specification (DS)				1

Level of Detail:

1. Section/paragraph level of detail.

2. Section level detail.

3. An ORG is required for each program/runstream per reference D-S7. The ORG takes the place of the Computer Operations Manual (OM) at HQ AFGWC.

NOTE 1: Project Manager has the option of waiving this document by sending a letter to the section tasked

NOTE 2: Current prologue requirements will suffice for new development projects costing 500 hours or less. The PM has the authority to request a separate MM if he so desires.

Documentation Standards

D-S5, SOFTWARE LISTINGS

1. **Applicability.**
 - a. This standard is to be used for any Review/Audit Standard which requires "software listings" to be reviewed.
 - b. No software listings are required for deletions.
2. **References:** N/A.
3. **Requirements.** This standard describes and defines what goes into the software listings. The following items are required:
 - a. Map (link, SEGLDR, LDR) listing if applicable.
 - b. If only a partial retrofit, or no retrofit, of standards was performed:
 - (1) Listing of source code from latest baselined version of the software (i.e., "old code").
 - (2) Listing of source code from proposed software (i.e., "new code"), with line numbers starting with line number incremented by 1, compiled if applicable.
 - (3) DOWN processor listing. Use "aDOWN, L new, old".
 - c. If full retrofit of the standards is performed, or if this is new software: listing of source code from proposed software, compiled if applicable.
 - d. Use compiler options necessary to provide a cross-reference and non-standard extension usage listing, if applicable.
 - e. All legitimate processor/compiler warnings and errors not eliminated during the development or maintenance process will be documented and explained in the System Deficiencies section of the Test Analysis Report (see note).

NOTE: Warnings and errors associated with software may have an adverse impact on the system

Documentation Standards

D-S6, INFORM MESSAGE

1. **Applicability:** All subsystems and INFORM messages.
2. **Reference:** N/A.
3. **Requirements:** All subsystems will contain an INFORM message.
 - a. INFORM messages will contain only one record (one line). The record will start with the OSCL indicating a comment. It will contain the subsystem name, subsystem version (see below), and the subsystem build date. Additional information can be included. ON UNISYS 1100 series systems, the following format will be used:
UNISYS 1100 series format:
@ . *** subsystem name *** VERSION yy-mm dd mm yy
Unisys 1100 series example:
@ . *** DS *** VERSION 89-02 03 JAN 89
 - b. The subsystem version will consist of the two digit year, a dash, and the build number. The build number starts at #1 at the beginning of each year, and is incremented by one with each implemented version (e.g., 89-01, 89-02...). All implemented versions (including fall-forwards, EOTs, and emergencies) will be incremented.
 - c. On Unisys 1100 series systems, INFORM messages will be in an element. This element will have no version name. The element name may be INFORM or FFINFO, where FF is the family name.
4. **Notes:**
 - a. An INFORM message is the means of displaying which version of the subsystem is implemented.
 - b. EOTs which pass QAA/CCA can stay on-line without a new version number.

Documentation Standards

D-S7, OPERATOR'S REFERENCE GUIDE (ORG)

1. **Applicability:**
 - a. All production runstreams and TIP transactions must have a current ORG on file with HQ AFGWC/SYC/Hardware Tech Service
 - b. All subsequent changes to the software that cause changes to the ORG require that the ORG be updated by the responsible programmer(s).
 - c. HQ AFGWC/SYC/Hardware Tech Services can waive this standard.
2. **Reference:** N/A.
3. **Requirements:**
 - a. ORGs will comply with the format found in D-S7-A1.
 - b. Prior to QAA, the programmer will take a printed copy and a diskette containing the ORG to HQ AFGWC/SYC/Hardware Tech Services for approval. HQ AFGWC/SYC/Hardware Tech Services will indicate approval on the Form 28 and the hardcopy. The programmer will then turn in the hardcopy of the ORG and the Form 28 with the QAA package.
4. **NOTES:**
 - a. The ORG replaces the Operator's Manual (OM) at AFGWC.
 - b. The ORG shell (D-S7-A1) is available in ENABLE format from HQ AFGWC/SYC/Hardware Tech Services.
 - c. The Implementation Manager will identify on the Form 13 the file name of the associated ORG.
 - d. HQ AFGWC/SYS/Quality Assurance will retain the hardcopy of the ORG.

D-S7-AI, EXAMPLE OF OPERATOR'S REFERENCE GUIDE

```

*****
*                                     *
*               OPERATOR'S REFERENCE GUIDE               *
*                                     *
*               FOR (name)                               *
*                                     *
*****

```

1. DATE ORG WAS UPDATED dd mmm yy
=====

ORG WAS UPDATED BY programmer/team/section
=====

2. RUNSTREAM (RUNID) NAME
=====

The runstream name as it appears on the operator's console, which is the same as the six character run-id field from the @RUN card

OR the TIP transaction name.

3. DESCRIPTION
=====

A brief description of what the runstream or TIP transaction does and how it does it.

4. TAPE(S) TO BE MOUNTED
=====

List all the number(s)/name(s) of any tapes that are used by the runstream. Identify if internally Labeled ("J" option on @ASG?) and the density (U9V, U9S, . . .). Also say whether the program will read, write, or both.

5. OPERATING INSTRUCTIONS
=====

A. START PROCEDURES
=====

-- HOW TO START THE RUNSTREAM
=====

The command to start the runstream, if the runstream is normally auto-started, say what program starts it and the manual restart key-in.

For TIP transactions, give the commands to initialize and to deactivate.

-- SETC OPTIONS
=====

Include any SETC options along with their impact.

B. CONSOLE MESSAGES
=====

-- List ALL messages that go to the operator's console.

-- Say whether or not an operator response is required.

-- List acceptable operator responses and their effects.

C. RUN PRIORITY
=====

From the @RUN card.

D. AFGWC COMPUTER SYSTEMS
=====

List which computer systems the program can be executed on.

6. ERROR STOPS AND RECOVERY PROCEDURES

=====

Include a brief explanation of the response to all. possible errors, including
 aborts.

7. PROGRAM INFORMATION

=====

A. ABSOLUTE ELEMENTS AND SUBSYSTEMS

=====

List all elements and their associated subsystems that are added or executed by the program (runstream).

B. INPUT

=====

-- PREDECESSORS

=====

List the RUNID(s) of any required predecessors

-- INPUT DATABASES

=====

List by LITERAL and LABEL

-- OTHER REQUIRED INPUT FILES

=====

C. OUTPUT

=====

-- OUTPUT DATABASE

=====

List by LITERAL and LABEL

-- OTHER OUTPUT FILES

=====

8. MAINTENANCE PROGRAMMER

Section and team (if applicable). Do NOT list anyone's name, or any phone numbers - HP AFGUC/SYC/Hardware Tech Services will use the recall rosters for that information.

9. DISPOSITION OF OUTPUT

=====

A. DUTY SECTION/GIN

=====

Name of duty section or bin ID where print should be delivered, if any. (Error print?)

B. OUTPUT DEVICES

=====

Include other output devices (Digital Communications Terminal (DCT), Keyboard Send/Receive (KSR), etc.)

C. BULLETIN HEADINGS

=====

D. MANOP NUMBERS

List manual operator (MANOP) numbers if sent over the Automated Weather Network (AWN) or other communication time.

CONFIGURATION ~~MANAGEMENT~~ STANDARDS

M-S1 - Elements Added or Executed

20 Apr 90

Requires elements Added or executed to be in a controlled subsystem

MS2 - Subsystem and Production Program File Naming Standards

20 Apr 90

Defines file naming standards for new production systems or new program files.

Software Configuration Management Standards

M-S1, ELEMENTS ADDED OR EXECUTED

1. Applicability:

- a. Any new software must follow this standard.
- b. Any software which is completely overhauled (greater than 40% change per General Standard #1) must follow this standard.

2. Reference. N/A.

3. Requirements. Any element added or executed by production software must be in a controlled subsystem. This does not apply to data elements or Team Chief Options. Data elements contain no operating system control Language statements. Team Chief Options are Located in P*CA3VX and only start other runstreams.

4. Notes. A controlled subsystem is one which goes through formal CCA.

Software Configuration Management Standards

M-S2, SUBSYSTEM AND PRODUCTION PROGRAM FILE NAMING STANDARDS

1. Applicability:

- a. Any new subsystem or new production program file on a UNISYS 1100 series system must follow this standard.
- b. Any new software on a UNISYS 1100 series system must be considered as a candidate to go into a file which follows this standard.

2. Reference: N/A.

3. Requirements:

- a. Subsystems will be composed of a single AFGUC software family when practical. In this standard, "sy" denotes subsystem and "FF" denotes family.

(1) If the subsystem contains an entire family and only that family, replace "sy" with the two-letter family designation from SDSL.

(2) If the subsystem contains only one subfamily, replace "sy" with five letters, two for the family and three for the subfamily name (e.g., DTHYP for DT/HYP).

(3) Otherwise coordinate the naming with HQ AFGWC/SYC/Hardware Tech Services.

- b. As a minimum a subsystem will consist of the following files:

(1) sy*ECL. Contains the ECL (add-elts, etc.) and the INFORM message (see D-S6). Specific compile and map elements may reside in this file.

(2) sy*ABS. Contains the absolutes (and/or the collected relocatables) for the subsystem

(3) sy*SRC. Contains the source code for the subsystem

- c. A subsystem in addition to the files listed in 3.b, may also contain the following files:

(1) sy*REL. Contains the relocatables for the subsystem

(2) FF*LIBSRC. Contains the source code which is commonly used (i.e., by more than one program) within the family, but is unique to that family.

(3) FF*LIBREL. Contains the relocatables which are output from compiling the source code in FF*LIBSRC.

(4) sy*STORE. Is available for use by the SSM. If used, it will contain software that is awaiting implementation.

(5) sy*RESTORE. Is available for use by the IM. If used, it will contain copies of the previous elements which were modified for the current version.

(6) sy*BUILD. Contains the support software (e.g., build runstreams, test data elements, test drives) necessary to build and test the subsystem's production software.

- d. These elements in the short key-in file are also part of the subsystem

(1) The runstream that loads the subsystem from tape.

(2) Six-line runstreams that add ECL elements in sy*ECL. Start elements (runstreams) consist of: @RUN, @HDS and @ADD.

4. NOTES: The above subsystem files may reside on disk in the following manner:

(1) sy*ECL. - Logical production and logical development systems.

(2) sy*ABS. - Logical production and logical development systems.

(3) sy*SRC. - Logical development systems only.

(4) sy*REL. - Logical development systems only.

(5) FF*LIBSRC. - Logical development systems only.

(6) FF*LIBREL. - Logical development systems only.

(7) sy*STORE. - Logical development systems only.

(8) sy*RESTORE. - Logical development system only.

(9) sy*BUILD. - Logical development system only.

PROGRAMMING STANDARDS

P-S1 - Indentation	1 Jul 89
Requires indentation of FORTRAN code blocks.	
P-S2 - Console Message/Response	1 Jul 89
Defines messages to and responses for system console operator.	
P-S3 - Variable and Constant Names and Declarations	1 Apr 91
Defines convention for declaring variables.	
P-S4 - Statement Numbering	1 Apr 91
Requires that FORTRAN statements within each routine be in ascending order, incremented by 10 or more.	
P-S5 - ASSIGN Statement	1 Jul 89
Forbids use of ASSIGN.	
P-S6 - Arithmetic IF Statement	1 Jul 89
Forbids use of arithmetic IF.	
P-S7 - Error Checking	1 Jul 89
Requires that input data be edited for validity.	
P-S8 - GO TO Statements	1 Jul 89
Mandates minimal use of unconditional and computed GO TO statements.	
P-S9 - Library Processes	1 Apr 91
Mandates use of library in lieu of user-written processes.	
P-S10 - Use of Parentheses	1 Jul 89
Mandates use of parentheses to avoid ambiguity in FORTRAN routines.	
P-S11 - FORTRAN Extensions	1 Jul 89
Mandates avoidance of compiler extensions.	
P-S12 - JUMP Command in Runstreams and ADD Elements	20 Apr 90
Restricts use of the JUMP command.	
P-S13 - Length of Arithmetic Assignment Statements	20 Apr 90
Restricts the use of continuation lines for arithmetic assignment statements	
P-S14 - Internal Organization of Routines	1 Apr 91
Specifies a uniform organization in each routine for the prologue declarative, format, and executable statements.	
P-S15 - Routine Size	1 Apr 91
Specifies a maximum routine size of 180 lines or less of executable code.	
P-S16 - Assigning Tape Files	1 Jul 89
Defines the format and content of tape assignment statements.	
P-S17 - Use of the INCLUDE statement and the FORTRAN PROC	1 Apr 91
Defines the proper use of PROCEDURES and INCLUDE statements.	
P-S18 - Common Blocks	1 Apr 91
Defines the use of COMMON blocks.	

Programming Standards

P-S1, INDENTATION

1. Applicability:

a. In accordance with Standard G-S1. The G-S1 Retrofit Table also applies individually to any changed DO Loop or block IF.

b. Applies to any added DO loop or block IF.

2. References. N/A.

3. Requirements:

a. Each level of subordination will be indented a consistent number of spaces (two to six).

b. DO loops, nested DO Loops, and block IF statements occurring within the range of a DO Loop will be indented.

c. DO loops occurring within the range of a block IF statement will be indented and all the statements within the range of the loop will be indented additionally.

d. Examples showing minimum acceptable indention are attached. Additional indention may be inserted to enhance readability at the programmer's option.

4. Note. The purpose of this standard is to improve the readability of computer programs by use of a uniform structure which reflects the logic of the program

P-W-AI, INDENTATION EXAMPLES

Examples of DO statement and block-IF statement indentation follow. Note that the block-IF statement of FORTRAN 77 can only be simulated with Unisys FORTRAN 5.

FORTRAN 77

```

ELVS=ELVS+ZIP
DO 200 I=1,10
  ..ARRAY(I,21)=ELVS
  ..DO 100 J=1,20
    ....DO ARRAY(I,J)=VECTOR(J,I)
  ..CONTINUE
100 CONTINUE
200 CONTINUE

```

FORTRAN 5

The same

```

ELVS=ELVS+ZIP
IF(ELVS.EQ.5)THEN
  ..CALL ROUTN
  ..ZIP=ZIP+1
END IF

```

```

ELVS=ELVS+ZIP
IF(.NOT.(ELVS.EQ.5))GO TO 100
  ..CALL ROUTN
  ..ZIP=ZIP+1
100 CONTINUE

```

```

POINTR=POINTR+1
IF(POINTR.GT.5)THEN
  ..CALL SUBTOT(DOLLARS,CENTS)
  ..CALL NEXTREC(BFR,LSTREC)
  ..POINTR=0
ELSE
  ..DOLLARS=DOLLARS+BFR(1,POINTR)
  ..CENTS=CENTS+BFR(2,POINTR)
END IF

```

```

POINTR=POINTR+1
IF(.NOT.(POINTR.GT.5))GO TO 100
  ..CALL SUBTOT(DOLLARS,CENTS)
  ..CALL NEXTREC(BFR,LSTREC)
  ..POINTR=0
  GO TO 200
100 CONTINUE
  DOLLARS=DOLLARS&+BFR(1,POINTR)
  ..CENTS=&+BFR(2,POINTR)
200 CONTINUE

```

```

IF(ELVS.EQ.5)THEN
  ..IF(TIP.EQ.3)THEN
    ....CALL ROUTA
  ..ELSE
    ..CALL ROUTB
  ::END IF
ELSE
  ..CALL ROUTC
END IF

```

```

IF(.NOT.(ELVS.EQ.5))GO TO 300
  ..IF(.NOT.(TIP.EQ.3))GO TO 100
    ..CALL ROUTA
    ::GO TO 200
  100 ...CALL ROUTB
  200 ..CONTINUE
  GO TO 400
  300 CONTINUE
  ..CALL ROUTC
  400 CONTINUE

```

```

IF(ELVS.EQ.A)THEN
  ..CALL ROUTA
ELSE IF (ELVS.EQ.B)THEN
  ..CALL ROUTB
ELSE IF (ELVS.EQ.C)THEN
  ..CALL ROUTC
ELSE
  ..ZIP=SPAM
END IF

```

```

IF(.NOT.(ELVS.EQ.A))GO TO 100
  ..CALL ROUTA
  GO TO 400
  100 IF(.NOT.(ELVS.EQ.B))GO TO 200
    ..CALL ROUTB
    GO TO 400
  200 IF(.NOT.(ELVS.EQ.C))GO TO 300
    ..CALL ROUTC
    GO TO 400
  300 CONTINUE
  ..ZIP=SPAM
  400 CONTINUE

```

DO 100 I=1,10	DO 200 I=1,10
..SUM=SUM+A(I)	..SUM=SUM+A(I)
..IF(A(I).LT.0)THEN	..IF(.NOT.(A(I).LT.0))GO TO 100
....SUM=SUM-(2*A(I))SUM=SUM-(2*A(I))
..END IF	100 ..CONTINUE
100 CONTINUE	200 CONTINUE

ELVS=B	ELVS=B
IF(ELVS.EQ.ZIP)THEN	IF(.NOT.(ELVS.EQ.ZIP))GO TO 200
.. DO 100 I-1,10	.. DO 100 I-1,10
....ELVS=ELVS+A(I)ELVS=ELVS+A(I)
100 ..CONTINUE	100 ..CONTINUE
ELSE	GO TO 300
..ELVS=A(9)	200 CONTINUE
END IF	..ELVS=A(9)
	300 CONTINUE

Programming Standards

P-S2, CONSOLE MESSAGE/RESPONSE

1. Applicability:

- a. In accordance with Standard G-S1.
- b. Applies only to any new or modified individual message software on Unisys 1100 systems, except that paragraph 3a is universal.

2. References:

- a. Unisys 1100 Operator Reference Manual.
- b. AFGWC FLASH and AFLASH Systems Documentation.

3. Requirements:

- a. Any new message routed to a system console from a production program must be coordinated with SYC/Hardware Tech Services prior to implementation.
- b. Individual message lines cannot exceed operational character limits.
- c. Message text should be clear, concise, and direct. Information messages will be kept to a minimum.
- d. Standard operator responses (responses and formats need not be explained in the message):
 - A - Yes, go ahead, continue processing.
 - X - Terminate run.
- e. Non-standard responses and formats will be explained in the message itself. If the operator enters a response other than standard and other than those listed in the message, the program should repeat the original message (give the operator a second chance).
- f. All messages and possible responses must be explained in the program's Operator's Reference Guide (ORG).
- g. Programs will not contain the FORTRAN PAUSE "message." It causes a program to abort unless the operator gives the non-standard response "S".
- h. Messages directing an operator to complete an unusual or complicated "fix" will refer to the program's ORG rather than use many console messages to make the same point. The following example demonstrates the proper format for such a message: RUNID*SEE ORG ERROR FIX 23.

4. Notes. Standard and non-standard console message examples:

RUNID*ENTER DTG IN DATES FORMAT (bad)

RUNID*ERROR MONTH/DAY/YEAR MM/DD/YY OR X (good)

RUNID*BAD RETURN FROM NFIELD. CONTINUE? (bad)

RUNID*ABNORMAL STATUS DETECTED. ANS A TO GO, X TO ABORT (good)

RUNID*THIS PROGRAM EXTRACTS DATA FROM

RUNID*THE GLOBAL APPLICATIONS

RUNID*DATA BASE. WHEN YOU ARE READY TO

RUNID*PROCESS REPLY WITH TRANSMIT. (bad--delete information messages)

RUNID*ENTER DAY DD

RUNID*ENTER HOUR HH

RUNID*ENTER YEAR YY (bad)

RUNID*ENTER DAY/HOUR/YEAR DD/HH/YY OR X (good)

Programming Standards

P-S3, VARIABLE AND CONSTANT NAMES AND DECLARATIONS

1. Applicability:

- a. If the routine has all of its variables and constants described in the prologue, and retrofit of the prologue is not required by D-S2, paragraphs 3d(1) and 3d(6) do not apply.
- b. Any variable or constant added to a routine during modification will be declared in a manner consistent with the rest of the routine.
- c. In accordance with Standard G-S1.

2. References. None.

3. Requirements:

- a. All names will be consistent throughout the entire process. That is, a data element will not change names across routine boundaries. However, there is an exception to this rule: A routine which is called with more than one set of arguments. For instance, in the following example, NORMAL is exempt:

```
REAL FUNCTION NORMAL (VAL, LOW HIGH)
REAL VAL
REAL LOW
REAL HIGH
NORMAL = MAX (MIN(VAL,HIGH),LOW)
RETURN
END

REAL VALUE
REAL NEWVAL
REAL CLOUD(64,64,15)

VALUE = NORMAL (NEWVAL,0.,100.)
CLOUD(I,J,K) = NORMAL (CLOUD(I,J,K),10.,99.)
```

- b. All names will be chosen in meaningful fashion, so that the name or mnemonic used gives a reasonable clue to contents of the data element. For example, "SPEED" is better than "X1234" for wind speed.

- c. Routine listings will include a cross reference listing and a Listing of variable types. See D-S5.

d. Declarations:

- (1) All variables, constants and parameters must be explicitly declared. IMPLICIT NONE will be used if the compiler supports it.
- (2) Place all variable declarations for a common block's variables together above the common statement.
- (3) Place all declarations for the parameters together above the parameter statements. Place the parameter statements together before the other type statements, unless Logic or syntax demand otherwise.
- (4) Except for paragraphs (2) and (3) above, declarations will be grouped in one of three ways:
 - (a) Alphabetically by name.
 - (b) Alphabetically by type.
 - (c) Functional grouping.
- (5) There will be only one declaration per line.
- (6) All variables and constants will be described in at least one of the following ways (the method used within any routine must be consistent):
 - (a) In in-line comments. Use "!" (space, exclamation mark, space) if the compiler supports it. Use " @ " (space, at sign, space) with Unisys's FOR and FTN compilers. If the description continues beyond the end of the line, the next line will use an in-line comment aligned under the first.
 - (b) In a Data Dictionary, Design Dictionary, or another CASE supported representation; or in a Data Specification, Maintenance Manual, or some other document. If the variable or constant can be easily referenced by its name, one reference in the prologue is sufficient. Otherwise, each variable or constant must have a precise reference to the description on the declaration.
- (7) DIMENSION statements will not be used. Dimensioning will be done as part of the variable declaration.

- e. DATA statements will be grouped together. They will follow the type statements and precede the executable statements.

- f. DO Loop control variables and DO loop control expressions will be Integer.

4. Note: This standard is intended to improve readability and maintainability.

Programing Standards

P-S4, STATEMENT NUMBERING

1. **Applicability.** In accordance with Standard G-S1.
2. **Reference.** Kernighan, B.W., and Plauger, P.J. 1978: The Elements of Programing Style. McGraw Hill, 168pp
3. **Requirements:**
 - a. Statement numbers must be in ascending order within each routine.
 - b. Statement numbers will initially be incremented by at least 10.
4. **Notes:**
 - a. Reason for 3a: Minimize time required to locate a given statement number.
 - b. Reason for 3b: Allows numbered statements to be inserted when the routine is modified and still meet 3a.

Programming Standards

P-S5, **ASSIGN STATEMENT**

1. **Applicability.** In accordance with Standard G-S1.
2. **Reference.** Yourdon.
3. **Requirement.** Do not use the ASSIGN statement; a CALL/RETURN is better.
4. **Notes.** The ASSIGN statement makes programs difficult to follow and debug.

Programing Standards

P-S6, ARITHMETIC IF STATEMENT

1. **Applicability.** Per Standard G-S1.
2. **Reference.** Kernighan and Plauger.
3. **Requirement.** Do not use the arithmetic IF statement.
4. **Notes:**
 - a. Good programming practice minimizes use of branching statements.
 - b. Instead of the arithmetic IF, use the IF THEN ELSE construct (FORTRAN 77).

Programming Standards

P-S7, ERROR CHECKING

1. **Applicability:** In accordance with Standard G-S1.
2. **Reference:** N/A.
3. **Requirements:**
 - a. Edit input data for validity and plausibility. This does not apply to data passed from one module or program to another.
 - b. Check I/O statements for errors. Use the "IOSTAT=" or the "ERR=" clauses on all 'READ' statements, and on any other appropriate I/O statements.
 - (1) Use the 'IOSTAT=' clause then check the returned value to determine if an error has occurred. ANSI Standard FORTRAN 77 requires that the 'IOSTAT=' clause return an integer value. Checking the returned value will be limited to the following conditions as defined in the ANSI Standard:
 - (a) Less than zero: An End-of-File was encountered and no error exists.
 - (b) Zero: No End-of-File was encountered and no error exists.
 - (c) Greater than zero: An error has occurred.
 - (2) The 'ERR=' and 'END=' clauses may only be used where a 'GOTO' would be allowed (see P-S8).
4. **Notes:**
 - a. Checking input data for validity and plausibility helps make a program robust (i.e., resistant to failure). Although it is not mandated that data passed between modules be checked, it is encouraged that bounds checking be applied where critical.
 - b. The 'END=' and 'ERR=' clauses are implicit 'GOTO' statements. As such, they are subject to the standards governing the use of 'GOTO' statements.
 - c. Other than those described above, there is no intrinsic meaning to numbers returned by 'IOSTAT='. Because vendors assign their own meanings to the returned value, checking for a given error code introduces vendor-dependent code.

Programing Standards

P-S8, GO TO STATEMENTS

1. **Applicability.** In accordance with Standard G-S1.

2. **Reference.** N/A.

3. **Requirements:**

a. **Avoid unnecessary branches.** Minimize the use of the GO TO statement. In FORTRAN 77, the GO TO statement may be used only in the following ways:

(1) **Do-Until structures,** where a section of code is repeated until a logical condition is met. For example:

```
C -----
C Read until an End-of-File is detected
C -----

10 CONTINUE
  READ (UNIT=U1, FMT=CFMT, IOSTAT=STAT, ERR=20, END=20) VAR
  GO TO 10
20 CONTINUE
```

----- OR -----

```
C -----
C Compute until a certain condition is met
C -----

10 CONTINUE
  ---sequential code---
  IF (logical expression) GO TO 10
```

(2) **Do-While structures,** where a section of code will be checked before execution and will be executed until a logical condition is met. For example:

```
C -----
C FORTRAN 77 Do-While construct.
C -----

10 CONTINUE
  IF (logical expression) THEN
    ---sequential code---
    GO TO 10
  END IF
```

(3) **Exit from a loop nested inside an outer loop or Do-Until or Do-While construct.** Being well nested and desiring an exit requires a GO TO. This is the only case where a GO TO should be employed for jumping forward in a routine. For example:

```
C -----
C Exit from the inner loop when a condition is met.
C -----

DO 70 I = 1, IMAX
  ---sequential code---
  IF (logical expression) THEN
    ---sequential code---
    DO 60 J = 1, JMAX
      ---sequential code---
60    CONTINUE
      IF (logical expression) GO TO 80
    END IF
70 CONTINUE

C -----
C Exit here from Loop when special condition is met.
C -----

80 CONTINUE
```

b. In FORTRAN 5 do not use a computed GO TO statement except in the following instances:

- (1) A CASE structure is simulated within a code block, and
- (2) Range of the variable is tested.

c. In FORTRAN 77, do not use a computed GO TO statement_

4. Note. See Attachment 1 for examples of CASE structures.

P-S8-A1, CASE STRUCTURE EXAMPLES

Example of a CASE structure in FORTRAN V:

```

C      -----
C      CASE ENTRY . . . COMMENTS EXPLAINING CASES LABEL
C      -----

      IF ((INDX.GE.1).AND.(INDX.LE.5)) GO TO 10

C      -----
C      INVALID VALUES OF INDX
C      -----

      ---code for invalid values---
      GO TO 60
10    GO TO (20,30,40,20,50), INDX

C      -----
C      CASES 1 AND 4 COMMENTS EXPLAINING CASES 1 AND 4
C      -----

20    ---code for cases 1 and 4---
      GO TO 60

C      -----
C      CASE 2 COMMENTS EXPLAINING CASE 2
C      -----

30    ---code for case 2---
      GO TO 60

C      -----
C      CASE 3 COMMENTS EXPLAINING CASE 3
C      -----

40    ---code for case 3---
      GO TO 60

C      -----
C      CASE 5 COMMENTS EXPLAINING CASE 5
C      -----

50    ---code for case 5---
60    CONTINUE

C      -----
C      ENDCASE
C      -----

```

Example of a CASE structure in FORTRAN 77:

```

C      -----
C      CASE ENTRY COMMENTS EXPLAINING CASES
C      -----

      IF ((INDX.EQ.1).OR.(INDX.EQ.4)) THEN

C      -----
C      CASES 1 AND 4 COMMENTS EXPLAINING CASES 1 AND 4
C      -----

      ---code for cases 1 and 4---
      ELSE IF (INDX.EQ.2) THEN

C      -----
C      CASE 2 COMMENTS EXPLAINING CASE 2
C      -----

      ---code for case 2---
      ELSE IF (INDX.EQ.3) THEN

C      -----
C      CASE 3 COMMENTS EXPLAINING CASE 4
C      -----

      ---code for case 3---
      ELSE IF (INDX.EQ.5) THEN

C      -----
C      CASE 5 COMMENTS EXPLAINING CASE 5
C      -----

      ---code for case 5---
      ELSE

C      -----
C      INVALID VALUES OF INDX
C      -----

      ---code for invalid cases---
      END IF

```

Programming Standards**P-S9, LIBRARY PROCESSES**

1. **Applicability.** In accordance with Standard G-S1.
2. **References.** FLASH documentation, AFLASH documentation, Unisys ASCII FORTRAN Reference Manual, American National Standard Programming Language FORTRAN X3.9-1978.
3. **Requirement.** Use intrinsic functions and library processes (e.g., SPRT, COPY, SIN) when available.
4. **Note:**
 - a. Use of intrinsic functions and Library processes saves computer and programming time.
 - b. Some libraries are: TIP\$*TIPREL\$, TIP\$*TIPLIB\$, AFGWC*USERSREL, AFGWC*ASCIIREL, AFGWC*ASCIIREL-65K, SYSS*RLIB\$, and SYSS*ALT\$.

Programing Standards

P-S10, USE OF PARENTHESES

1. **Applicability.** In accordance with Standard G-S1.
2. **References.** None.
3. **Requirements.** Use parentheses to avoid ambiguity, for example:

 Use $A = ((PI/4) * ((RADIUS(I) ** 4) * PI)) / 16 + ((PI ** 2) * 25)$ (good),

 Not $A = PI/4 * RADIUS(I) ** 4 * PI / 16 + PI ** 2 * 25$ (more confusing).

 Use IF(((PI/4)*DIAM**2).LT.((X*AREA)/(12*H**2))) .AND.
 \$ ((VOLUME**(1./3)).GT.((SQRT(AREA*50))*H))
 \$ SURFACE = 02
4. **Note.** Clarity is a most important quality of any computer program

Programming Standards**P-S11, FORTRAN EXTENSIONS**

- 1. Applicability.** In accordance with Standard G-S1.
- 2. References.** American National Standards Institute FORTRAN X3.9-1978.
- 3. Requirements:**
 - a. Do not use extensions to FORTRAN 77 except:**
 - (1) if a construct is explicitly allowed by one of the AFGWC Software Standards, or
 - (2) if BITS, SBITS, VIRTUAL, MOVWDS, COMPILER PAGESIZE options, octal data, or FIELDATA are the constructs in question, or
 - (3) if the extension is necessary for a TIP (MCB) interface or
 - (4) if SYR has given explicit approval.
 - b. Any use of an extension must be explicitly documented in the prologue unless an automated listing of all extensions is provided for the QAA and filed in the SDF.**
- 4. Notes:**
 - a. FORTRAN 77 is a minimal standard. It requires that compatible compilers properly process standard source statements. It does not forbid extensions in compatible compilers. Therefore, portability demands that source programs conform to the standard, i.e., that compiler extensions be avoided.**
 - b. If available with the compiler used, the option to flag extensions will be turned on in the listings provided for QAA.**

Programing Standards

P-S12, USE OF THE @JUMP COMMAND IN RUNSTREAMS AND ADD ELEMENTS

1. **Applicability.** In accordance with Standard G-S1, for runstreams and ADD elements only.
2. **References.** N/A.
3. **Requirements.** Minimize use of the @JUMP command.
 - a. @JUMP statements will transfer control to a label only. The @JUMP will not be used to go forward a specified number of lines.
 - Valid: @JUMP NEXT
 - Invalid: @JUMP 5
 - b. When a label is used, no EXEC command will be appended to the Label.
 - Valid: @NEXT:
@ASG,T FILE.
 - Invalid: @NEXT:ASG,T FILE.
4. **Notes:**
 - a. Using the @JUMP command places program Logic into runstreams, making them emulate programs. The project programmer/analyst should determine if the purpose of the project would be better served by an absolute rather than a runstream. Large runstreams, over time, become difficult to analyze, redesign, and maintain. They also become prone to logic errors as they change. Runstreams also are not portable as is FORTRAN, and should be kept as short as possible.
 - b. When the @JUMP command is used to move forward a specified number of lines, and a maintenance programmer adds a new command (without changing the number of lines to be jumped), the @JUMP may transfer control to the wrong block of commands. The practice of using a label with the @JUMP command eliminates this problem.
 - c. Separation of the label from the block of commands improves the modifiability of that block and decreases the probability of logic errors. It also improves the readability of the runstream or ADD element.

Programming Standards

P-S13, LENGTH OF ARITHMETIC ASSIGNMENT STATEMENTS

1. **Applicability.** In accordance with Standard G-S1.
2. **Reference.** None.
3. **Requirements.** Minimize continuation of arithmetic assignment statements from one line to the next.
 - a. **When a continuation is used:**
 - (1) The continuation character will be an '&'.
 - (2) The continued code will begin in the same column as the first line of the arithmetic expression.
 - b. **Arithmetic assignment statements may be continued when:**
 - (1) needed for Cray vectorization.
 - (2) array indices consist of multiple variables.
 - (3) the assignment statement has been indented one or more times.
4. **Example:**

```

C -----
C  LOOP THRU NSMT NUMBER OF TIMES TO SMOOTH DATA
C -----
C
C      DO 500 KK = 1, NSMT
C          DO 200 J = 2, JMAXM1
C              DO 100 I = 2, IMAXM1
C                  B(I,J) = A(I,J) + XA(I,J) * (A(I+1,J) + A(I-1,J) +
&                      A(I,J+1) + A(I,J-1) - 4.0 * A(I,J)) +
&                      XB(I,J) * (A(I+1,J+1) + A(I+1,J-1) +
&                      A(I-1,J+1) + A(I-1,J-1) - 4.0 * A(I,J))
100      CONTINUE
200      CONTINUE
500      CONTINUE

```

Programming Standards

P-S14, INTERNAL ORGANIZATION OF ROUTINES

1. **Applicability.** In accordance with Standard G-S1. This standard does not apply to Assembly Language.
2. **References.** Software Quality Assurance, Thomas McCabe and Associates, McCabe Press, 1980.
3. **Requirements.**
 - a. Single entry and exit points. Routines are entered at the top, and exited at the physical bottom. Only one RETURN statement per routine is allowed. Only one STOP statement per program is allowed. The STOP statement will be located in the main routine.
 - b. All FORMAT statements will be grouped together, and placed in one of the locations shown in paragraph d.
 - c. If the language used supports it, there will be a statement for the routine ID.
 - d. The internal elements of routines, if present, will be sequenced as follows:
 - (1) Routine ID. (Examples: PROGRAM SUBROUTINE, FUNCTION, or @RUN.)
 - (2) Process Narrative, if required (see D-S1).
 - (3) Routine prologue (see D-S2).
 - (4) Non-executable code (see P-S3).
 - (5) FORMAT statements (one possible location).
 - (6) Code.
 - (7) STOP or RETURN statement.
 - (8) FORMAT statements (other possible location).
 - (9) END statement
 - e. Runstreams are an exception to 3.d. They will begin with a RUN card, then a HDG card, then the prologue.
4. **Notes:**
 - a. Standard organization of software modules contributes to maintainability, testability, flexibility, portability, and reusability.
 - b. Statement numbering will follow the scheme given in Standard P-S4.

Programming Standards

P-S15, ROUTINE SIZE

1. **Applicability:**
 - a. In accordance with Standard G-S1.
 - b. Not applicable to Cray software in those instances where larger routine sizes enhance vectorization and significantly increase speed of execution.
2. **Reference.** Reliable Software Through Composite Design, Glenford J. Myers.
3. **Requirements:**
 - a. Routine size is limited to 180 lines of executable code (i.e., excluding prologue, variable declarations, FORMAT statements, etc.).
 - b. A routine will contain only one function.
4. **Notes:**
 - a. Small routines are easier to read, increasing maintainability.
 - b. Single function routines reduce code complexity, shorten debug time, and make software modification simpler. Problems are more quickly isolated to a particular routine.

Programming Standards

P-S16, ASSIGNING TAPE FILES

1. **Applicability:** In accordance with Standard G-S'1, for software on Unisys 1100 systems.

2. **References:**

- a. Series 1100 EXEC Reference, UP-4144.30, 3.7.1.2.
- b. Series 1100 Hardware/Software Mini-Reference, UP-7824, 1.4.

3. **Requirements:** When assigning tapes, use the format described below.

@ASG,options filename,type, reel,,ring-indicator

a. **Options.** This option affects options T and J. Other options may also be used.

(1) Use "T" unless catalogued tape files are required (e.g., SECURE runs).

(2) "J" may only be used when at least one of the following conditions apply. Document the applicable condition(s) in the prologue.

(a) Tapes will be used on a system that doesn't allow labeled tapes (e.g., System 1/4).

(b) Tapes will be used by outside agencies or on non-Unisys hardware.

(c) The user has a documented requirement for unlabeled tapes.

b. **Filename.** Acceptable formats are:

(1) A character string that identifies either the tape's contents or the software that uses the tape.

(2) The filename or FORTRAN unit number expected by absolutes in the runstream

c. **Type.** Specify the type (U9V, U9S, etc). Do not use "T".

d. **Reel.** A S-digit numeric field consisting of the external tape number with leading zeros.

(1) When a specific tape number is not required, this field may be omitted.

(2) When the tape has no external number (takers, spews, FAX tapes, etc.) and a FORTRAN unit number is used for the filename, an identifying character string (e.g., Q1SPEW, TK1) may be used in the reel field.

e. **Ring indicator.** Either "RING" or "NORING".

4. **Notes:**

a. Using Labeled tapes provides a safeguard against using the wrong tape. If a specific reel number is called for, that is the only one the EXEC will accept.

b. The EXEC includes leading zeroes in the reel number, so a standard length is required. For example, the EXEC sees 12 and 00012 as different numbers.

Programming Standards

P-97, USE OF INCLUDE STATEMENT AND THE FORTRAN PROC

1. Applicability:
 - a. In accordance with G-S1, excluding MSM
 - b. New processes, common blocks, and processes using logical database views will meet this standard.
2. References. FORTRAN 77 Programmer Reference Manual, System Utilities Programmer Reference Manual.
3. Requirements:
 - a. Use of the INCLUDE for common blocks and logical database views.
 - (1) Use the INCLUDE statement and PROC for common block declarations.
 - (2) Limit the number of common blocks per PROC as noted in P-S18.
 - (3) Have only one logical database view per PROC and vice versa.
 - b. In keeping with the variable names and declarations standard (P-S3), data declarations may be used in a PROC.
 - c. Do not place non-related data declarations in a PROC.
 - d. Do not place executable code in a PROC.
 - e. The INCLUDE is not intended for simple insertion of prologues or comments.
 - f. PROCs will be used only when they will be included in two or more routines.
 - g. Each PROC will contain a prologue as required by D-S2.
 - h. PROC documentation can be accomplished in several ways:
 - (1) PROCs can be "compiled" with an @PDP,FL.
 - (2) Use the LIST option on the INCLUDE statement.
 - (3) Explicitly defining common variables in the process narrative (see D-S1) to fulfill the documentation requirement.

Programming Standards**P-S18, Common Blocks**

1. **Applicability:**
 - a. In accordance with G-S1.
 - b. New processes, and processes using logical database views will meet this standard.
2. **References.** FORTRAN 77 Programmer Reference Manual, System Utilities Programmer Reference Manual.
3. **Requirements:**
 - a. Stank (unlabeled) common blocks will not be used.
 - b. Common blocks will be defined and declared within a FORTRAN PROC and merged into the program via the INCLUDE compiler command.
 - c. Character and numeric data variables will not be intermingled within a single common block as required by P-S11.
 - d. Only one common block definition will be allowed within a single proc, with one exception: a common block containing only character data may be in the same PROC as a common block containing non-character data.
 - e. All variables will be explicitly typed and conform to variable naming and declaration standards as stated in P-S3.
 - f. Data declarations related to a specific common block will precede its common statement as stated in P-S3.
 - g. Data contained within a common block will be functionally related to each other and arranged for understandability, so that:
 1. All data elements within a common block form a logical subset of data, e.g., a logical database view.
 2. Common block PROCs will only be included in routines that reference at least 50% of the variables within the common block. The exception to this will be when common blocks are used as an "interface" with other processes external to your process, e.g. pre-defined database views, preexistent data structures, etc.
 - h. Data items that do not meet the above criteria (e.g.) will be passed as subroutine arguments, unless a strong logical dependency exists.
 1. Common blocks will be used when they will be included into two or more routines.
4. **Note.** The intent of this standard is to encourage responsible use of the common block. It is suggested the use of common blocks be minimized, since over usage and poorly structured common blocks result in a high degree of coupling and make debugging time consuming and difficult.

REVIEW/AUDIT STANDARDS

R-S1 - General Requirements	1 Apr 91
Addresses requirements for all reviews and audits.	
R-S2 - System Requirements Review (SRR)	1 Apr 91
Specifies elements of an effective SRR.	
R-S3 - System Design Review (SDR)	1 Apr 91
Specifies elements of an effective SDR.	
R-S4 - Preliminary Design Review (PDR)	1 Apr 91
Specifies elements of an effective PDR.	
R-S5 - Critical Design Review (CDR)	1 Apr 91
Specifies elements of an effective CDR.	
R-S6 - Product Verification Review (PVR)	1 Apr 91
Specifies elements of an effective PVR.	
R-S7 - System Validation Review (SVR)	1 Apr 91
Specifies elements of an effective SVR.	
R-S8 - Functional Configuration Audit (FCA)	1 Jul 89
Specifies elements of an effective FCA.	
R-S9 - Physical Configuration Audit (PCA)	1 Jul 89
Specifies elements of an effective PCA.	
R-S10 - Quality Assurance Audit (QAA)	1 Apr 91
Specifies elements of an effective QAA.	
R-S11 - Configuration Control Audit (CCA)	1 Jul 89
Specifies elements of an effective CCA.	

Review/Audit Standards

R-S1, GENERAL REQUIREMENTS

1. Applicability. All software projects.

2. References:

- a. DOD-7935.1-STD, Automated Data System (ADS) Documentation Standards.
- b. AFR 205-16, Automated Data Processing (ADP) Security Policy, Procedures, and Responsibilities
- c. AFR 700-4, Vol I, Information Systems Program Management.
- d. AFGWC 700-1, Communication-Computer System Project and Configuration Management.
- e. AFGWC 700-2, Computer Software Standards.
- f. AFGWC 171-5, Subsystem Implementation.
- g. Software Standard D-S4, Minimum Documentation Requirements.

3. Requirements:

a. This standard addresses requirements that apply to all reviews and audits. The other review/audit standards identify requirements for specific reviews/audits according to project cost.

b. Objectives:

(1) Software reviews ensure visibility and technical understanding and verify that the system will satisfy requirements.

(2) Software reviews ensure that security is considered and that security requirements are met.

(3) Software audits verify achievement of development requirements and identify a product configuration.

c. Responsibilities:

(1) Project Manager (PM). The single individual with the full authority and responsibility for managing a project. There is only one project manager for a given project; however, a project manager may manage more than one project.

(2) All Participants. Prepare for and actively participate in reviews, ensure that the project will meet user requirements, review documentation and provide written comments to the PM, inform the PM if more time is needed to prepare for reviews, and help ensure effective developer-user communication.

(3) User. The PM, when necessary, will represent the user. Ensure that requirements are understood by the developer; review documents emphasizing completeness, accuracy, and readability; ensure all requirements have been satisfactorily met.

(4) HQ AFGWC/SYR. Provide guidance on configuration management and software standards.

(5) HQ AFGWC/SYS/Quality Assurance. Advise on quality assurance considerations, review documents emphasizing testability and test procedures, conduct the QAA, ensure standards compliance.

(6) Programmer. Help the PM prepare for reviews, attend all reviews and audits.

d. The following items will be addressed at all reviews in addition to specific requirements listed in individual reviews.

(1) Security requirements (ref 2b).

(2) Requirements of previously omitted reviews.

(3) Minutes of previous reviews, with emphasis on action items.

(4) Documentation reviewed, approved, or baselined at a previous review, as required by questions about or changes to that document.

(5) Configuration Management (CM) records, including AFGWC Forms IO.

e. Review process:

(1) See Standards R-S2 through R-S7. Use the HQ AFGWC Software Review Checklist (R-S1-A1).

(2) Reviews are required for all development or maintenance efforts which require substantive functional software design or redesign.

(3) The PM will announce reviews and make required documentation available to participants at least 2 weeks before the scheduled date. In any case, reschedule if attendees are not prepared in time.

(4) The PM will make an agenda to guide the meeting.

(5) The PM will invite to reviews the user, HQ AFGWC/SYR, HQ AFGWC/SYS/Quality Assurance, and other functional work areas as appropriate.

(6) The PM will conduct the review.

(7) Compromises for the best solutions are made, and baselines/documents are approved by consensus. The user and configuration management and quality assurance personnel as well as the developers, must be satisfied. The project/program manager is responsible for project direction and process execution. ALL decisions concerning conflicts with AFGWC 700-2 will be handled by HQ AFGWC/SYR.

(8) The PM will write minutes and include action items with OPRs and suspenses indicated.

f. Physical and Functional Configuration Audits (PCA, FCA):

(1) See Standards R-S8 and R-S9.

(2) The PM will inform HQ AFGWC/SYR of the anticipated audit requirement as soon as possible.

(3) The PM will chair the audit and write minutes. HQ AFGWC/SYS/Quality Assurance will help review documentation and code.

g. Quality Assurance Audits (QAAs) and Configuration Control Audits (CCAs) are conducted by HQ AFGWC/SYS/Quality Assurance per Review/Audit standard R-S10 and R-S11 respectively. QAAs and CCAs are normally required before all implementations.

h. When new requirements or additional time to complete a project causes the review/audit requirements to change, the PM will determine whether the new review/audit requirements will be applied to the project or not. The new requirements will be requested via AFGWC Form 10, Software Change Request.

4. Notes:

- a. R-S1-A2 specifies the reviews/audits required for each project as a function of estimated cost
- b. R-S1-A3 summarizes the desired status of various documentation in relation to review and audit points.

1 2

HP AFGWC SOFTWARE REVIEW CHECKLIST (R-S1-A1)

QAA

OCT 1988

YES NO

1. Pre-Review:

- a. Identify review and audit requirements early in the project?
- b. Announce review/audit 2 weeks before scheduled?
- c. Documentation ready and available to players 2 weeks ahead?
- d. Project ready for this review or audit?
- e. Participants ready (know project, reviewed documentation, etc.)?
- f. Review AFGWC Software Standards R-S1 through R-S11, as applicable?
- g. Identify general goals (see also Standard R-S1)?
 [Meet user requirements;
 SIP Objectives of portability and maintainability;
 Security (AFR 205-16);
 Quality software meeting AFGWC Software Standards;
 Identify and track problems.]
- h. Identify specific goals of this review/audit (Standards R-S2 through R-S11)?

2. During the Review

- a. Agenda prepared (handout, viewgraph, or verbal)?
- b. Quick reminder of general and specific goals?
- c. Include objectives of omitted reviews?
- d. Cover minutes of previous reviews, especially action items?
- e. Is the user represented?
- f. Is HQ AFGWC/SYR represented?
- g. Review "bogged down" in technical problem solving or detailed or administrative review of documentation?
- h. Get agreement that review objectives were met?
- i. Assign OPRs and suspenses on action items?

3. Documentation:

- a. Form appropriate (handwritten, draft, final)?
 - b. Format correct (DOD 7935.1-STD, AFGWC 171-5)?
 - c. Content OK (accurate, complete)?
 - d. Documentation understandable?
 - e. Editorial problems (punctuation, grammar, etc.)?
-

4. Post-Review

- a. Write minutes?
[Include: objective, significant questions and answers,
deviations, recommendations, decisions, action items, attendees.]
- b. Follow up action items?

R-SI-AZ, REQUIRED REVIEW/AUDITS BASED ON ESTIMATED PROJECT COST

REVIEW REQUIRED	PROJECT COST			
	0-500wh	501-1728wh	1729wh-17280wh	More than 17280wh
System Requirement Review (SRR)			X	X
System Design Review (SDR)			X	X
Preliminary Design Review (PDR)				X
Critical Design Review (CDR)			X	X
Product Verification Review (PVR)			X	X
System Validation Review (SVR)		X	X	X
Functional Config. Audit (FCA)				X
Physical Config. Audit (PCA)				X
Quality Assurance Audit (QAA)	X	X	X	X
Configuration Control Audit (CCA)	X	X	X	X

Example: For a project costing 1000whs, an SVR, QAA, and CCA will need to be conducted.

Review/Audit Standards**R-S2, SYSTEM REQUIREMENTS REVIEW (SRR)**

1. **Applicability.** Software projects estimated to cost more than 1728 wh.
2. **References.** See Review/Audit Standard #1.
3. **Requirements:**
 - a. **Definition.** The SRR validates the user requirements and determines the initial direction of the design effort. With approval of the Functional Description (FD) at the SRR, the functional baseline is established.
 - b. **Objectives:**
 - (1) Approve the functional baseline.
 - (2) Review security aspects, ensure inclusion in the functional baseline.
 - (3) Ensure clear understanding of the requirements.
 - (4) Determine direction of the initial design effort.
 - (5) Identify/review project milestones, present project strategy.
 - (6) Identify critical or high-risk items.
 - c. **Timing.** Conduct this review at the end of the conceptual phase after a significant portion of the functional requirements are defined and before establishing the functional baseline.
 - d. **Documents reviewed:**
 - (1) Information Systems Program Plan (ISPP).
 - (2) Functional Description (FD). [Draft]
4. **Notes.** The functional baseline consists of the FD.

Review/Audit Standards**R-S3, SYSTEM DESIGN REVIEW (SDR)**

1. **Applicability.** Software projects estimated to cost more than 1728 wh.
2. **References.** See Review/Audit Standard #1.
3. **Requirements:**
 - a. **Definition.** The SDR validates the total system requirements used in producing design specifications.
 - b. **Objectives:**
 - (1) Approve the allocated baseline.
 - (2) Review security specifications, identify assumptions and constraints, determine requirements for redoing the risk analysis or revising the System Test and Evaluation (ST&E) Plan.
 - (3) Reach technical understanding of requirements, provide technical direction to participants.
 - (4) Ensure the allocated requirements (high-level design) implement the functional requirements in the FD.
 - (5) Ensure technical risks are identified, ranked, and avoided, or trade-offs are made.
 - c. **Timing.** Conduct at the end of the definition phase when the design has progressed to the point where system characteristics and design approach are defined and CPCIs are identified.
 - d. **Documents reviewed:**
 - (1) Functional Description (FD). [Final]
 - (2) Data Requirements Document (RD). [Draft]
 - (3) System Specifications (SS). [Draft]
4. **Note:** May combine this with the Preliminary Design Review (PDR) of a single-CPCI system.

Review/Audit Standards**R-S4, PRELIMINARY DESIGN REVIEW (PDR)**

1. **Applicability.** Software projects estimated to cost over 17280 work-hours.
2. **References.** See Review/Audit Standard #1.
3. **Requirements:**
 - a. **Definition.** The PDR allows a review of the preliminary design for each Computer Program Configuration Item (CPCI) of the system. For systems of a single CPCI, the PDR may be combined with the System Design Review (SDR).
 - b. **Objectives:**
 - (1) Review security specifications and System Test and Evaluation (ST&E) Plans, as required.
 - (2) Review the design progress and design technical adequacy. Consider interfaces, storage, databases, timing, human factors.
 - (3) Ensure design compatibility with the Subsystem Specification (SS) and other CPCIs.
 - (4) Consider testability of the selected design.
 - c. **Timing.** Conduct this review early in the development phase, after the SS is baselined and before the detailed design.
 - d. **Documents reviewed:**
 - (1) Previously delivered, as required.
 - (2) Database Specification (DS). [Draft]
 - (3) Test Plan (PT) [Draft], Less test procedures. The PT will only contain black box test cases.
4. **Note.** Conduct on each CPCI or several selected CPCIs.

Review/Audit Standards**R-SS, CRITICAL DESIGN REVIEW (CDR)**

1. **Applicability.** Software projects estimated to cost more than 1728 wh.
2. **References.** See Review/Audit Standard #1.
3. **Requirements:**
 - a. **Definition.** The CDR is conducted to validate the detailed design against the requirements, prior to the system or Computer Program Configuration Item (CPCI) being coded.
 - b. **Objectives:**
 - (1) Review security specifications and System Test and Evaluation (ST&E) Plans, as required.
 - (2) Review security aspects of the Operators Reference Guide (ORG) and Users Manual (UM).
 - (3) Ensure the detailed design is consistent with the functional specifications and allocated baseline.
 - (4) Ensure the design is fully defined to permit coding.
 - c. **Timing.** Conduct this review when the detailed design is complete, before coding this part of the system.
 - d. **Documents reviewed:**
 - (1) Previously delivered, as required.
 - (2) Data Requirement Document (RD). [Final]
 - (3) System Specification (SS). [Final]
 - (4) Program Specification (PS). [Draft]
 - (5) Users and Maintenance Manuals (UM, MM) and Operators Reference Guide (ORG), all draft.
 - (6) Test Plan (PT) covering black box testing only. [Draft]
4. **Note.** May be done incrementally, in order to release certain functions or CPCIs for coding.

Review/Audit Standards**R-S6, PRODUCT VERIFICATION REVIEW (PVR)**

1. **Applicability.** Software projects estimated to cost more than 1728 work hours.
2. **References.** See Review/Audit Standard #1.
3. **Requirements:**
 - a. **Definition.** The PVR validates the Computer Program Configuration Item (CPCI) or system as being ready for formal testing.
 - b. **Objectives:**
 - (1) Approve the product baseline.
 - (2) Ensure compliance with security specifications.
 - (3) Ensure development is complete and the CPCI(s) is(are) ready for validation (system) testing.
 - c. **Timing.** Conduct this review at the end of the development phase after the preliminary audit(s). If the PVR is required and an EOT will be accomplished, the PVR must be held prior to the EOT being put on line.
 - d. **Documents reviewed:**
 - (1) Previously delivered, as required.
 - (2) Preliminary audit findings.
 - (3) Test Plan (PT) (final).
 - (4) Software listings.
4. **Note:** The product baseline consists of the Program Specifications (PS), Database Specifications (DS), and software listings.

Review/Audit Standards

R-S7, SYSTEM VALIDATION REVIEW (SVR)

1. Applicability. Software projects estimated to cost more than 500 work-hours.
2. References. See Review/Audit Standard #1
3. Requirements:
 - a. Definition. The SVR validates the tested system against the requirements of the Functional Description (FD) and System/Subsystem Specification (SS).
 - b. Objectives:
 - (1) Ensure system adequately addresses security requirements.
 - (2) Ensure the system through validation (system) testing, performs as stated in the FD and SS, and that it meets customer requirements.
 - (3) Ensure that each element is traceable to FD requirements.
 - (4) Resolve loose ends for customer acceptance.
 - c. Timing. Conduct this review at the end of the testing phase after completing validation testing and the final audit(s).
 - d. Documents reviewed:
 - (1) Previously delivered, as required.
 - (2) Database Specification (DS). [Final]
 - (3) Program Specification (PS). [Final]
 - (4) User's Manual (UM). [Final]
 - (5) Maintenance Manual (MM). [Final]
 - (6) Test Analysis Report (RT).
 - (7) Final audit findings.
 - (8) Software listings.

Review/Audit Standards**R-S8, FUNCTIONAL CONFIGURATION AUDIT (FCA)**

- 1. Applicability.** Software projects estimated to cost more than 17280 wh.
- 2. References.** See Review/Audit Standard #1.
- 3. Requirements:**
 - a. Objectives.** Validate satisfactory software development. Ensure that its performance meets specifications and customer requirements.
 - b. Timing.** Conduct the preliminary audit late in the development phase, before the Product Verification Review (PVR). Conduct the final audit late in the testing phase before the System Validation Review (SVR).
 - c. Documents reviewed:**
 - (1) Program Spec (PS).
 - (2) Subsystem Specification (SS).
 - (3) Test Plan (PT).
 - (4) Test Analysis Report (TAR) at final audit.
 - (5) Software listings.
- 4. Notes:**
 - a.** This audit may be combined with the Physical Configuration Audit (PCA).
 - b.** The preliminary audit includes the bulk of the effort. The final audit concentrates on changes made during testing.

Review/Audit Standards**R-S9, PHYSICAL CONFIGURATION AUDIT (PCA)**

1. **Applicability.** Software projects estimated to cost more than 17280 wh.
2. **References.** See Review/Audit Standard #1.
3. **Requirements:**
 - a. **Objectives:**
 - (1) Ensure that the coded version of the software agrees with its technical documentation.
 - (2) Establish the product baseline.
 - b. **Timing.** Conduct the preliminary audit late in the development phase, before the Product Verification Review (PVR). Conduct the final audit late in the testing phase before the System Validation Review (SVR).
 - c. **Documents reviewed:**
 - (1) Program Specification (PS).
 - (2) Users Manual, Maintenance Manual, Operators Reference Guide (UM, MM ORG).
 - (3) System Specification (SS).
 - (4) Test Plan (PT).
 - (5) Test Analysis Report (RT) at final audit.
 - (6) Software listings.
4. **Notes:**
 - a. This audit may be combined with the Functional Configuration Audit (FCA).
 - b. The preliminary audit includes the bulk of the effort. The final audit concentrates on changes made during testing.

Review/Audit Standards

R-S10, QUALITY ASSURANCE AUDIT (QAA)

1. **Applicability.** All software development, modification, and maintenance projects regardless of estimated cost.
2. **References:**
 - a. See Review/Audit Standard #1.
 - b. AFGUCR 171-6, Software Library Operations.
3. **Requirements.** A QAA is completed for all software projects. General considerations are summarized in Review/Audit Standard #1.
 - a. **Objective.** The QAA evaluates system design, code, testing, documentation, compliance with standards, and coordination. The audit goal is to minimize the possibility that an implementation will have an adverse impact on AFGWC operational systems.
 - b. **Auditor.** The auditor is the HQ AFGWC/SYS/Quality Assurance representative who performs the code and documentation inspection. The auditor provides a report of audit which is presented at the out-brief.
 - c. **Attendance.** The out-brief is attended by the program/project manager and the responsible programmer. Responsible HQ AFGWC/SYS functional area chief attendance is required for AFGUC in-house projects or at program/project manager request.
 - d. **Timing.** HQ AFGWC/SYS/Quality Assurance conducts this audit after all testing and coordination is complete. The audit out-brief will occur at the end of the audit process.
 - (1) The programmer will coordinate a desired QAA date with HQ AFGWC/SYS/Quality Assurance and turn in a complete package at least two duty days prior to that date. Major or large audits may take longer to complete.
 - (2) QAA packages for software implemented as an emergency also require approval of the emergency implementation, per SY OI 171-6. They do not require a Test Plan (PT); however, the Test Analysis Report (RT) should address in general terms what and how testing was done and results.
 - e. **Documents reviewed:**
 - (1) AFGWC Form 28.
 - (2) Applicable Configuration Management forms as required by AFGUCR 700-1.
 - (3) HQ AFGWC/SYS waivers, if applicable.
 - (4) Life Cycle documentation, as applicable (see D-S4):
 - (a) Operator's Reference Guide (ORG).
 - (b) Test Plan (PT), pre-approved by HQ AFGWC/SYS/Quality Assurance.
 - (c) Test Analysis Report (CRT).
 - (d) Test output.
 - (5) Software Development Folder (SDF).
 - (6) Software listings (see D-S5).
4. **Note:** The QAA will also include elements of other reviews that were not previously completed.

Review/Audit Standards

R-S11, CONFIGURATION CONTROL AUDIT (CCA)

1. **Applicability.** Families being released for implementation on AFGWC computer systems.
2. **References:**
 - a. See Review/Audit Standard #1.
 - b. AFGWCR 171-6, Software Library Operations.
3. **Requirements.** A CCA is completed prior to HQ AFGWC/SYS/Quality Assurance release of a subsystem for implementation.
 - a. **Objective.** The CCA is a review of the family build intended to ensure that only approved changes are made.
 - b. **Auditor.** The auditor is the HQ AFGWC/SYS/Quality Assurance representative who performs the documentation inspection.
 - c. **Notification.** Project/program manager and implementation manager will be notified of results at completion of CCA.
 - d. **Timing.** HQ AFGWC/SYS/Quality Assurance conducts this audit after the family tapes are built. The implementation manager will coordinate a desired CCA date with HQ AFGWC/SYS/Quality Assurance and turn in a complete package at least one day prior to that date.
 - e. **Documents audited:**
 - (1) AFGWC Form 13.
 - (2) Approved AFGUC Form(s) 28.
 - (3) Listing of the old and new Inform ELTs.
 - (4) Listing of the family tapes. An FLIST of each tape with added/changed elements highlighted, or a highlighted FLIST of one tape and some means to verify that the others are duplicates of it and are readable. (e.g., building one tape, copying it to a second tape, copying the second to a third, etc., and FLISTing the last tape).
4. **Notes:**
 - a. A listing of the tape build showing the deletions and copies is also desirable.
 - b. After HQ AFGWC/SYS/Quality Assurance releases the family, the implementation Manager coordinates with HQ AFGWC/SYC and the affected programmers on implementation times.

OPERATIONAL TASK STANDARDS

O-S1 - Operational Task Documentation

1 Jul 89

Defines the proper method for including comments and documentation lines into operational tasks (OTs).

O-S2 - Filename Conventions

1 Jul 89

Defines the file management procedures and naming conventions to be used when writing OTs.

Review/Audit Standards**R-S11, CONFIGURATION CONTROL AUDIT (CCA)**

1. **Applicability.** Families being released for implementation on AFGWC computer systems.
2. **References:**
 - a. See Review/Audit Standard PI.
 - b. AFGWC 171-6, Software Library Operations.
3. **Requirements.** A CCA is completed prior to HQ AFGWC/SYS/Quality Assurance release of a subsystem for implementation.
 - a. **Objective.** The CCA is a review of the family build intended to ensure that only approved changes are made.
 - b. **Auditor.** The auditor is the HQ AFGWC/SYS/Quality Assurance representative who performs the documentation inspection.
 - c. **Notification.** Project/program manager and implementation manager will be notified of results at completion of CCA.
 - d. **Timing.** HQ AFGWC/SYS/Quality Assurance conducts this audit after the family tapes are built. The implementation manager will coordinate a desired CCA date with HQ AFGWC/SYS/Quality Assurance and turn in a complete package at least one day prior to that date.
 - e. **Documents audited:**
 - (1) AFGWC Form 13.
 - (2) Approved AFGWC Form(s) 28.
 - (3) Listing of the old and new Inform ELTs.
 - (4) Listing of the family tapes. An FLIST of each tape with added/changed elements highlighted, or a highlighted FLIST of one tape and some means to verify that the others are duplicates of it and are readable. (e.g., building one tape, copying it to a second tape, copying the second to a third, etc., and FLISTing the last tape).
4. **Notes:**
 - a. A Listing of the tape build showing the deletions and copies is also desirable.
 - b. After HQ AFGWC/SYS/Quality Assurance releases the family, the implementation Manager coordinates with HQ AFGWC/SYC and the affected programmers on implementation times.

OPERATIONAL TASK STANDARDS

O-S1 - Operational Task Documentation

1 Jul 89

Defines the proper method for including comments and documentation lines into operational tasks (OTs).

O-S2 - Filename Conventions

1 Jul 89

Defines the file management procedures and naming conventions to be used when writing OTs.

--

Operational Task Standards

0-SI, OPERATIONAL TASK DOCUMENTATION

1. **Applicability:** All operational tasks (OTs) implemented within the Satellite Data Handling System or the AWDS Product Driver Subsystem as part of any AFGUC Operation Task System (OTS) and developed by AFGUC Task Developers or Harris MetApp Maintainers.
2. **References:**
 - a. AFGUC Operational Task System (AFGWC/OTS) Data Base Specification.
 - b. Computer Program User's Manual (CPUM) for the Satellite Data Handling System (SDHS), Volume VII, Operational Task Builder/Compiler, latest version;
 - c. CPUM for the SDHS, Volume VIII, Forecaster Command Language, latest version.
3. **Requirements:**
 - a. The Operational Task Compiler (OTC) allows an "unlimited" number of comment lines. These lines begin with an exclamation mark (!) in column 1, and the OTC ignores any such line it finds.
 - b. This document shows how to include these comments and documentation lines as the OTs are written. For the Subsystem File Management (SSF) procedures to work correctly, the format given in the following sections will be used--varying from these formats will yield unpredictable results!
 - c. The following list of requirements will be observed:
 - (1) No lines in the OT will exceed 79 characters in length.
 - (2) All prologue documentation will appear between the second line of the OT ("MH" or "OT") and the first \$\$\$SUBTASK.
 - (3) Block-type documentation lines will only be allowed in the SSVOCAB and SSCOMMAND sections.
 - (4) All comments and documentation lines to be printed in the final formatted document will begin with either "!!" or "!*" in the first two columns of the line.
 - (5) No !! or !* lines will come between a \$\$\$SUBTASK label line and the following SSCOMMAND line. The only entry allowed is the subtask name.
 - (6) Each vocabulary statement in a SSVOCAB section and each SSCOMMAND section (except for the top-level menu) will include one "!*" comment line that tells the purpose of that vocab or command sequence. These are the lines that will be extracted to produce the Task Structure listing.
 - d. The following two pages provide an example of how an OT will be documented. Note the entries "PURPOSE:", "DEVELOPER:", "UPDATES:", and "PROCESSING:". These will be included exactly as shown (with applicable information supplied).
4. **Notes for the attached OT Documentation Example:**
 - a. The date and version number in the last "UPDATES:" line and in the first line of the top-level menu MST always match.
 - b. The "FILES:" and "TASK STRUCTURE:" sections are NOT typed into the OT source documents. These are created dynamically on the MenuVax system by the SSF DOC procedure.
 - c. Use the long "!!!!!!" lines or tab-indent the subtask lines to separate the SUBTASKS for easier reading. Shorter lines may be used to set off the different sub-sections within each subtask.
 - d. One "!*" comment will be used ahead of each vocabulary statement in the SSVOCAB section and prior to the command strings in each SSCOMMAND section of all non-menu subtasks. Without these, there would be no reference to the menu choice in the Task Structure document.
 - e. Document all changes to an OT in the actual code where the changes are made. This applies to additions or deletions. Filename changes are exempt.

O-SI-AI, OPERATIONAL TASK DOCUMENTATION EXAMPLE

DBMAFO

OT

```
=====
!! PURPOSE: The purpose of the DBMAFO operational task is to allow
!! the Database Manager to monitor the reliability of the AFOS data
!! stored in the display database (DDB).
!!
!! DEVELOPER: TSgt McGinnes/WFMS
!!
!! UPDATES:
!! 01FEB88/1.00 - Initial Operational Version
!! 23APR88/2.00 - Added function to display 3D maps--Mr. John
!!               Green/Harris METAPP
!! 30APR88/2.01 - OTCR# 88025...Corrected bad CCM command--
!!               SMS Freimund/WFMS
!!
!! PROCESSING:
!!
!! Upon entering this operational task the operator will be able to
!! perform the following functions:
!!
!! 1. Exit the operational task
!!
!! 2. Retrieve a directory listing of the SDHS AFOS database
!!    and store the directory, and display the directory locally
!!    under a temporary filename.
!!
!! 3. . . .ETC... This documentation is what is referred to as the
```

```

=====
$$SUBTASK
DEMAFO
$$COMMAND
!-----
1 =  MN
!-----
$$MENU

                ** REVIEW AFOS PRODUCTS **                30APR88/2.01

1  RETURN TO THE MASTER MENU
2  RETRIEVE AND DISPLAY THE DATABASE DIRECTORY OF AFOS PRODUCTS
3  PRINT THE DATABASE DIRECTORY OF AFOS PRODUCTS
4  CLEAN UP

!-----+-----+
SSPROMPT      <-----+
ENTER SELECTION:  | NOTE: These lines are used to |
!-----+         | make the OTS easier to read. |
                <-----+         | They will be ignored by both |
                <-----+         | the OTC and the SSF. |
                +-----+
$$DEFAULT      <-----+
0              +-----+
!-----+         | NOTE: The DEFAULT is in Col 1 |
                +-----+
$$VOCAB
!* Return to Master Menu <-----+
01 = MN          | NOTE: These lines |
02 = LEV,AFODIR  | will be printed in |
!* Send Directory file to the printer <-----+ the Task Structure |
03 = SND,AFODIR,PTR,AFODIR | Diagram by use of |
!! *** 1.02 *** - OTCR# 88025 - Added Subtask | the SSF DOC tool. |
04 = LEV,AFOCLN  +-----+
!-----+
$$HELP
PLACE HELP TEXT OR SECOND PAGE OF MENU HERE
!-----+
$$SUBTASK
AFODIR
$$COMMAND
!* Retrieve/display the Database directory of AFOS products <-----+
= DEL,AFODIR
= DBI,AFODIR,AFO
= GDF,AFODIR
!-----+
$$SUBTASK
AFOCLN
$$COMMAND
!* Clean up for the review AFOS products menu <-----+
!! *** 1.02 *** - OTCR# 88205 - Changed to clean up unneeded files.
= DEL,AFODIR
= DEL,ZDTMP1
= CMD,DEL,|ZDFIL2
!! *** END CHG 1.02 ***
!-----+
$$END

```

Operational Task Standards

0-S2, FILENAME CONVENTIONS

1. Applicability:

a. All operational tasks (OTs) implemented within the Satellite Data Handling System or AWDs Product Driver Subsystem as part of any AFGWC Operational Task System and developed by any AFGWC Task Developers or contractor Maintainers.

b. All files stored in the Forecaster Working Data Base (FWDB) or the Local Working Storage (LWS) on the operational forecaster consoles.

2. References:

a. AFGWC Operational Task System (AFGWC/OTS) Data Base Specification.

b. Computer Program User's Manual (CPUM) for the Satellite Data Handling System (SDHS), Volume VII, Operational Task Builder/Compiler, latest version.

c. CPUM for the SDHS, Volume VIII, Forecaster Command Language, latest version.

3. Requirements: The AFGWC OTS employs several thousand data files. Most are temporary files in Forecaster Working Storage (FWS) used by a single OT subsystem. Many of them however, are stored in the Forecaster Working Data Base (FWDB) for reference and use by more than one subsystem. Tracking these files is a significant task both for OT developers and for AFGWC SDHS Data Base Administration personnel. This standard defines the file management procedures and naming conventions to be used when writing OTs.

a. Each OT developer will be responsible for maintaining the Subsystem I/O List(s) for the subsystem(s) being updated/created. This list will reside in a file called SSS.SIO (where SSS is the subsystem ID) located in the subsystem specification directory. The SSS.SIO file may contain three kinds of records as described in the following paragraphs:

(1) Input/Output/Temporary (IOT) file records in SSS.SIO identify files that are retrieved from or stored to SDHS Data Bases (DDB, RSDb, FWDB) and Forecaster Working Storage (FWS). Input, output, and temporary files are defined as follows:

(a) INPUT FILES are either files an OT retrieves, with a data retrieval command, from an SDHS data base, or files in local storage that a subsystem accesses from FWS which have been created or retrieved by another subsystem.

(b) OUTPUT FILES are either files an OT stores in the FWDB, or files that a subsystem creates and leaves in local storage for another subsystem to access.

NOTE: A file which a task retrieves from an SDHS database and leaves in FWS for another task to access does not have to be identified as both an input and output file of the retrieving task. This practice should be avoided as much as possible because it makes OTs heavily interdependent.

(c) TEMPORARY FILES are files created by one of the OTs of a given subsystem but not referenced by the of any other subsystem. There are two subcategories of temporary files: transitory and checkpointed. Transitory files are generated during editing, thresholding, or arithmetic operations and deleted as soon as they are used. It is recommended that "generic" file names be used for transitory files. Checkpointed files are those an OT creates not used by any other task; they are stored in FWDB to provide for quick recovery in the event of a system failure.

(2) Interface file records in SSS.SIO identify files transmitted to (Sent files) or received from (Received files) another subsystem. These files are used to interface one operational subsystem to other subsystem(s).

(3) Display file records in SSS.SIO identify files used by an OT to create Image Graphic Monitor (IGM) displays. They may be any type of file, e.g., image, vector, edited plot; however, they are identified in the SSS.SIO specifically as display files by using a unique Source/Destination ID (see below) to indicate which IGM and overlay plane is used.

(4) The records in the Subsystem SSS.SIO file will have the following format:

```
Columns: 1      2      3      4      5      6
1234567890123456789012345678901234567890123456789012345678
|          |          |          |          |          |
SUBTSK   T   SDI   PFN.   LFN.....
      ^^      ^^      ^^      |-----Max of 44 characters-----|
```

Field Contents

SUBTSK The operational task name, where SUB = the Subsystem ID

T File Type: I = Input O = Output T = Temporary (I/O LOC)
 S = Sent R = Received D = Display
 V = Temp Input from LOC U = Temp Output to LOC

SDI Source/Destination ID: The Location identifier for the source or destination of the file. The valid source and destination IDs are defined in paragraph 3.b.(5) below.

PFN **Physical Filename.** This is the name by which the file is identified in Forecaster Working storage or in the Forecaster Working Data Base. The A conventions used for FVDB filenames are defined in paragraph 3 b below and Appendix 1 to this standard.

In order to avoid conflicts among s&systems, ALL FVDB filenames will be assigned by SDHS Database Administration (DBA). Filenames for local FWS files will be assigned by the OT developer.

LFN **Logical Filename.** This is an abbreviated descriptive name of a file. The LFN can be up to 44 characters long. LFNs will conform to the conventions defined in paragraph 3.c below and Appendix 3 to this standard.

(5) The following table defines the legal identifiers for the sources and destinations of files:

Processors

sss Where sss = the Subsystem ID for another OT subsystem Use for files or messages sent directly to other consoles and for files received directly from other consoles.

LOC Local. forecaster working storage. (DO NOT use the local Subsystem ID. This will help distinguish internal and external console traffic).

APV APDS Products from the Forecaster Consoles.

EDS External Distribution Subsystem (Use for product definitions only).

uxx U01, U04, UB5, U06 - UNISYS logical systems.

RTS For WPP RTOS

ipp For display files where "i" is the IGM number and "pp" is the 2-digit overlay plane number (for vector files).

Data Bases

AFD DDB AFOS Product Data Base

CLP DDB Clipboard Data Base

DBA DDB Forecaster Working Data Base A

SBA DDB Forecaster Working Data Base A (SEC)

TRA DDB Forecaster Working Data Base A (SCI)

DEB DDB Forecaster Working Data Base B

SBB DDB Forecaster Working Data Base B (SEC)

TBB DDB Forecaster Working Data Base B (SCI)

GRD DDB Gridded Data Base

OTF DDB Operational Task Data Base

PLT DDB Plot Data Base

RDB DDB Reference Data Base (Lnd/Sea and GeoPol)

RSD Raw Satellite Data Base

SGD DDB Satellite Global Data Base

NOTE: DO NOT use 'DDB' as a source ID. It is NOT specific enough. You must use GRD, PLT, OTF, SCI, SEC, or AFD instead.

External Devices/Circuits

AFO	AFOS	NOTE: The following values will				
BGD	Big D (Satt Display Machine)	be used for SDHS-generated product				
DF1	ADFS1 (Auto Digital Fax Sys)	transmissions to AFOS:				
DF2	ADFS2		MAP	CENTER	PNT	ROTN
DF3	ADFS3	TYPE	SCALE	LAT	LON	FACT
ID5	ID50					
LF1	LF8801 (LaserFax)	N. HEM	1	82.5	-99.00	25
LF2	LF8802 "	N. AMR	2	47.0	-102.00	25
MTP	Mtg Tape	CONUS	4	38.5	-98.00	25
NSS	NESS					
PPL	Printer/Plotter					
PTR	Printer					

b. A physical filename is the name by which SDHS software knows the files in the FVDB or !VS. These filenames must be six characters long and must begin with a letter. The DBA maintains the file naming scheme. Use of local (FWS) filenames are determined by the task developer although they are encouraged to use the FWS convention when applicable. Appendix 1 shows the correct formats, codes, and abbreviations. Appendix 2 Lists the variables used to encode/decode the physical filenames.

c. A Logical filename is an abbreviated descriptive name of a file not to exceed 44 characters. All logical filenames MUST be prefixed with a file type identifier of which the first three characters are unique. The file type identifier is crucial to the operation of automated file management tools. Appendix 3 shows the proper format for encoding logical filenames, and Appendix 4 Lists the valid Logical file types.

Operational Task Standards

O-S2, FILENAME CONVENTIONS, APPENDIX 1

- A_c** - AFOS files (system and subsystem). Characters 2-4 will be the AFOS PIL; c = file cycle number.
- Blag*** - BOGUS files (system data), where l = level, a = geographical area, g = gridded data parameter (H=HGT, P=PRES) except for 'X' which indicates station deletions, and * = file format (V=VCT, T=TXT).
- C** - CLIPBOARD files (system and subsystem).
- Dd****** - FC-destined files from UNISYS systems, where d= DDB Filename Data Type code (see below) and **** = DDB Filename codes for day, hour, minute, and sequence number.
- | "d" | DDB Filename Data Type | Data Type |
|-----|---------------------------|-----------|
| A | Aircraft/CATONE plot file | SPL |
| B | Computer Flight Plan file | CFP |
| C | 15-Layer 3DNEPH file | NEF |
| D | Image Data | IMD |
| E | Data-to-Forecasters | DFC |
| F | RTOS messages | RTO |
| G | Print/Punch file | PPF |
| I | IPADS file | IPA |
- DIR** - Directory listings.
- E** -
- F** - CFP/CFT files (System Data).
- FKYsnn** - FUNCTION KEY DEFINITIONS (Fcstr Aid) - where s = the subsystem code and nn = any alphanumeric sequence number.
- Gnlghc** - GRID FILES, DATA (System Data) - where m = grid mesh projection, l = Level code, g = grid parameter code, h = forecast hours code, and c = file cycle number.
- GRImlg** - GRID FILES, STANDARD (Fcstr Aid) - where m = grid mesh, l = level and g = gridded data parameter
- Isxxxx** - IMAGE FILES, pseudo, composite (System Data) - where s = Subsystem ID and xxxx = any alphanumeric string.
- Jsxxxx** - PROGRAMMABLE CURSOR FILES, where s = Subsystem ID and xxxx = any alphanumeric string.
- LLxxx1** - LAT/LON projections stored in protected local working storage, xxx = PSS, PSN, etc.
- LPxxxx** - LOOKUP TABLE, Pseudo-Imagery (Fcstr Aids), where xxxx=any alphanumeric string.
- LTCTtn** - LOOKUP TABLES (Fcstr Aids), where CT = the Enhancement Curve used, t = Satellite Type, and n = any alphanumeric sequence number. *NOTE*: CT can also be "Zs" (where s = the subsystem code) for use with pseudo-image lookup tables. In this case, "tn" can be any two characters.
- LTcuto** - LOOKUP TABLE, Enhancement Curves (Fcstr Aids), where cu = the Enhancement Curve used, t = Satellite Type, and o = Enhancement Curve origin (G = AFGUC, N = NWS/NESDIS).
- M** - MAP FILES (Fcstr Aid).
- MAP** - Map Backgrounds (Fcstr Aid).
- MAPhnn** - Low-Level Routes (Fcstr Aid), where hnn = the route number
- MAPxy** - LORES GEO/POL and LND/SEA map backgrounds, where xx = NH, SH, TD, or TG and y = L (LndSea) or G (GeoPol).
- MPAAy** - HIRAS GEO/POL and LND/SEA map backgrounds, where AA = WMO Geographical Area "AA", 'n' is an octant code as shown in the Grid in Sub Area Code Table (0=entire area), and y = L (LndSea) or G (GeoPol).

MPxxsZ - A map background for Subsystem "s" where "xx" is any two alphanumeric characters

MWxxxx - WDS Low Level Route/Area Haps.

N*sxxx - ANIMATION FILE LISTS, where * = animation type (S for Satellite, V for Vector), s = Subsystem ID, and xxx = any three alphanumeric characters.

O*axxx - PLOT/PLOT VECTOR FILES (System Data), where * = plot type, a = geographic area, and xxx is coded as needed to describe the product.

"*" Plot Type

A Aircraft plot files
B Aircraft plot vector files
C CATONE plot files
D CATONE plot vector files
N RTNEPH files
R RAW Data files
S Surface Plot files
Surface Plot vector files
U Upper Air plot files
Upper Air plot vector files
W Severe Wk plot files
X Severe Wk plot vector files

P*nmzp - PRODUCT FILES (EDS Bound Data), * - Product type, nm = the file number (00...ZZ), z = product vector file contents, and p = the part number (for multi-part products).

Note: Use "L" for file contents on Vector Product Legend Templates.

"*" Product Type

T Teletype Product files (except Point Warnings)
V Vector Product files
W Point Warning TTY Product files

pprqff - POLAR SATELLITE IMAGERY SET FILE (SCI area only), where pp = the polar satellite identifier, q = the relative orbit identifier, q = the quarter orbit identifier, and tt = the file type designator.

R - RTOS MESSAGE TEMPLATES (Festr Aids)

R****x - RTOS TTY Bulletin Forms, where **** = TTAA of MANOP heading.

RPATEN - RTOS Point Analysis Template.

RPWxxx - RTOS Point Warning Bulletin Forms

S*ainc - SATELLITE IMAGE FILES, (System Data), where * = Satellite type, a = type, a = geographical area, i = image file contents, n = sequence or segment number, and c = file cycle number.

"*" Satellite Type Logical Filename Abbrevs.

DNSP (1st Sat, Ascending) (DM1 A)
B DNSP (1st Sat, Descending) (DM1 D)
C DNSP (2nd Sat, Ascending) (DM2 A)
DNSP (2nd Sat, Descending) (DM2 D)
GOES E (GOE)
GOES Landline (GOL)
O TIROS (N10 Ascending) LAC (TRx A LAC)
P TIROS (N10 Descending) LAC (TRx D LAC)
Q TIROS (N10 Ascending) GAC (TRx A GAC)
R TIROS (N10 Descending) GAC (TRx D GAC)
S SGDB (SDB)

Note: use "n" for area segment number (0=non-segment)

TIROS (N9 Ascending) LAC (TRx A LAC) On TIROS, x = 1
U TIROS (N9 Descending) LAC (TRx D LAC) or 2 for 1st or
TIROS (N9 Ascending) GAC (TRx A GAC) 2d satettite.
GOES W (GOW)
X TIROS (N9 Descending) GAC (TRx D GAC)

TEM*xx - PLOT MODEL TEMPLATES (Festr Aids), where * = plot type and xx = any alphanumeric string.

"*" Plot Type

A Aircraft plot files
C CATONE plot files
F Computer Flight Plans

G Gridded Data
 S Surface plot files
 U Upper Air plot files
 V Severe Weather plot files
 X Special purpose plot files

TPaxxb - Standard Topography files (Fcstr Aids), where a = geographical area, xx = SGDB grid box number, and b = topogeographicat sector (Numeric code b = 1000m interval; Alpha code b = 500m interval).

TPaaxx - Tropical Topography files (Fcstr Aids), where aa = geographical area and xx = Tropical geographical sector (Numeric code xx = 1000m interval; Alpha code xx = 500m interval).

NOTE: See Map Breakdown for Tropical Geographical sectors.

T*sxxx - TEXT FILES (Fcstr Aids/Tools), where * = Text File Type, s = subsystem code, and xxx = any alphanumeric string.

"*" Text File Type

S STATIC - Created and edited only by PMOs
 T TEMPLATES (Forms) - Created by PMOs; used by Forecasters
 X DYNAMIC - Text files edited by Forecasters

Vatvhc - VECTOR FILES (System Data), where a = geographical area, t = Level, v = vector file contents, h = forecast hours, and c = file cycle number.

W* - IPAM PC Subsystem static files (SCI area only).

x* - SAVDOX Bobus Subsystem static files (SCI area only).

Xntghc - ISOPLETH FILES (Standard), where m = grid mesh projection, L = Level code, g = grid parameter code, n = forecast hour code, and c = file cycle number.

Y* - WSP development use (SCI area only).

Yntyhc - ISOPLETH FILES (Non-Standard), where m = grid mesh projection, L = level code, y = grid parameter code, h = forecast hour code, and c = file cycle number.

Z* - Contractor use (SCI area only).

Zs**nn - SUBSYSTEM FILES (transitory), where s = subsystem code, nn = any alphanumeric string, and ** is the 2-char file type. (NOTE: The logical filename type will be "TMP".)

"**" File Type

G Gridded Data files
 GP Gridded Plot files
 I Image files, pseudo and composite
 OA Aircraft Plot files
 OB Aircraft Plot Vector files
 oc CATONE Plot files
 OD CATONE Plot Vector files
 OS Surface Plot files
 OT Surface Plot Vector files
 ou Upper Air Plot files
 ov Upper Air Plot Vector files
 ow Severe Wk Parameter Plot files
 ox Severe Wk Plot Vector files
 RT RTOS files (NOT TTY product files)
 SG Satellite data: SGDB
 SR Satellite data: RSDB
 TP Topography files
 V Vector files
 X Isopteth files
 Y Non-Standard Isoptethed files

Operational Task Standards

O-S2, FILENAME CONVENTIONS, APPENDIX 2

1. Summary of file name codes:

Code	ID	Code Name
	a/aa	Geographical Area
	b	Geographical Sub Area
	c	File Cycle
	ff	Polar Imagery Set file type designator
	g	Gridded Data Parameter
	h	Forecast Hours, Days, and Climate Months
		Satellite Image File Contents
	i	Atmospheric Level
	m	Grid Mesh
	n	Sequence number (any alphanumeric char.)
	pp	Polar satellite identifier
	q	Polar satellite quarter orbit identifier
		Polar satellite relative orbit identifier
	s	AFGWC/OTS Subsystem
	t	Satellite type code
	v	Vector File Contents
	y	Non-Standard Gridded Data Parameter code
	z	Product Vector File Contents

2. a/aa/b - Geographical Area/Sub Area Codes

Code Value	(a) Geographical Area	Code Value	(b) Geographical Sub Area
0			
1	Asian Theater CAST	1 (A)	Northeast
2	Asian Window (ASW)	2	East
3	CONUS Theater (UST)	3 (B)	Southeast
4	CONUS Window (USW)	4	South
5	European Theater (EUT)	5 (C)	Southwest
6	European Window (EUWO)	6	West
7	CONUS Severe Window	7 (D)	Northwest
8		8	North
9		9	Central
A	Atlantic, North (NT)		
B	Atlantic, South (ST)		
C	Central America/Caribbean		
D		HiRes Geo/LndSea Grid Octants	
E	Eastern Hemisphere (EH)	+-----+-----+-----+	
F	Full Disk (GOES)	7=NW 8=N 1=NE	
G		+-----+-----+-----+	
H		6=W 9=C 2=E	
I	Indian Ocean (IO)	+-----+-----+-----+	
J		5=SW 4=S 3=SE	
K		+-----+-----+-----+	
L			
M	Middle East (ME)	Topo	Geographical Sectors
N	Northern Hemisphere (NH)	+-----+-----+	
O		D A	
P	Pacific (PA)	+-----+-----+	
Q	Pacific	C B	
R		+-----+-----+	
s	Southern Hemisphere (SH)		
T	Tropical Belt (TROP)		
U	Tropical Theater - Eastern (TEH)		
V	Tropical Theater - Western (TWH)		
U	Western Hemisphere (WH)		

Code Value	(aa) Geographical Area	Code Value	(aa) Geographical Area
AE	South East Asia	FE	Far East
AO	West Africa	HW	Hawaiian Islands
AU	Australia	IO	Indian Ocean
AU	Near East	MG	Madagascar
CA	Caribbean Area	MM	Mediterranean Area
	and Central America	PS	South Pacific
CI	China	SA	South America
CR	Canary Islands	us	United States
EA	East Africa	ZA	South Africa

Western Hemisphere

[illegible]

4. c - File Cycle Number

The file cycle number is required only for cyclic files stored in FWOBB (i.e., for recovery or temporary back purposes). If the files are to be stored in FUOBA (the normal case for cyclic files), then encode "c" as zero, or indicated below for HIRAS and GSM products.

Code Value	Base time of file in hours relative to the current n-hour cycle					HIRAS & GSM
	1 hrly file	3 hrly file	6 hrly file	12 hrly file	24 hrly file	
0	-0	-0	-0	-0	-0	
	-1	-3	-6	-12	-24 (1PD)	
2	-2	-6	-12	-24	-48 (2PD)	
3	-3	-9	-18	-36	-72 (3PD)	
4	-4	-12	-24	-48	-96 (4PD)	
5	-5	-15	-30	-60	-120 (5PD)	
6	-6	-18	-36	-72	-144 (6PD)	
	-7	-21	-42	-84	-168 (7PD)	
8	-a	-24	-48	-96	-192 (8PD)	
9	-9	-27	-54	-108	-216 (9PD)	
A	-10	-30	-60	-120	-240 (10PD)	
B	-11	-33	-66			
C	-12	-36	-72			
D	-13	-39	-78			
E	-14	-42	-84			
F	-15	-45	-90			
G	-16	-48	-96			
H	-17					
	-18					
	-19					
K	-20					
	-21					
M	-22					
N	-23					
O	-24					
P						

NOTE: Cycle Code is used as the file identifier for SVR MVA outline Areas and for Tropical Specialized Support.

P - Z are unassigned yet

NOTES: (1) Code value '4' = File isoplethed at 4 mb intervals
Code value '8' = File isoplethed at 8 mb intervals
Code values 0-9 are used for Tropical Cyclone storms 0-9.

5. ff - Polar imagery set file type designator.

Code Value	Description
II	High resolution Infra-Red image
I4	Low resolution Infra-Red image
VI	High resolution Visual image
v4	Low resolution Visual image
L0	Ramp grid locations
MD	Satellite space land/sea boundaries
IT	Top Frame Infra-Red lookup table
IM	Middle Frame Infra-Red Lookup table
IB	Bottom Frame Infra-Red lookup table
VT	Top Frame Visual lookup table
VM	Middle Frame Visual lookup table
VB	Bottom Frame Visual lookup table

6. g - Grid Parameter Code

Code		NH	NH	NH	SH	SH	SH	TROP	HRS
Value	GADB	EGT	HLF	WHL	EGT	HLF	WHL	WHL	GSM
0		CDB		CDB1	CDB			CDB1	CONV
1		CDT	QPF1	CDT1	CDT			CDT1	PPW
2		CTAS	QPF2	CDB2	CTAS			CDB2	SM
3		TPC	QPF3	CDT2	TPC			CDT2	VSHR
4		CLAS	CAM		CLAS	CANT			
5		CLTS	CTP		CLTS				
6		CMS			CMS				
7		CMS			CMS				
8		CHAS			CHAS				
9		CHTS			CHTS				
A		AGE		(TADV)	AGE		(TADV)		
B			BDPD	(VADV)			(VADV)		(VADV)
C	(2)		(2)	(2)		(2)	(2)	(2)	(2)
D	DPD		DPD	DPD		DPD	DPD	DPD	DPD
E				EPT					
F		I(1)		(5)			(5)		(5)
H	HGT		HGT	HGT		HGT	HGT	HGT	HGT
I		ICE	ICE		ICE				
J									(6)
K	THK		THK	THK		THK	THK	THK	THK
L	DVL		DVL	DVL		DVL	DVL	DVL	DVL
M				MDVL			(MDVL)		
N	DPT	DPT	DPT	DPT	DPT	DPT	DPT	DPT	DPT
O			OMG						OMG
P	PRS		PRS	PRS		PRS	PRS	PRS	
Q	(3)		(3)	(3)		(3)	(3)	(3)	(3)
R									RH
s									PPP
T	TMP		TMP	TMP		TMP	TMP	TMP	TMP
U	(4)		(4)	(4)		(4)	(4)	(4)	(4)
V				VRT			VRT		VRT
W	u-v		u-v	u-v		u-v	u-v	u-v	u-v
X	DIR		DIR	DIR		DIR	DIR	DIR	STLN
Y	SPD		SPD	SPD		SPD	SPD	SPD	SPD
Z									

NOTES: (1) 'I' = Radar reflexivity
 (2) 'C' = Height change (height falls)
 (3) 'Q' = Pressure change
 (4) 'U' = Temperature change
 (5) 'F' = Height falls from Med Rng D-Value fields (MDVL)
 (6) 'J' = Height rises

7. h - Forecast Hours/Days and Climo Months Code

Code	Fcst Hrs/Days/	Code	Fcst Hrs/Days/	Code	Fcst Hrs/Days/
Value	Climo Mnth	Value	Climo Mnth	Value	Climo Mnth
0	0 hours	C	3 days/72 hrs	0	Month 1 (Jan)
	3 hours	D	4 days/96 hrs	P	Month 2 (Feb)
2	6 hours	E	5 days/120 hrs	Q	Month 3 (Mar)
3	9 hours	F	6 days/144 hrs	R	Month 4 (Apr)
4	12 hours	G	7 days/168 hrs	S	Month 5 (May)
5	15 hours	H	8 days/192 hrs		Month 6 (Jun)
6	18 hours	I	9 days/216 hrs	U	Month 7 (Jul)
3	21 hours		10 days/240 hrs	V	Month a (Aug)
a	30 hours	K	27 hours	W	Month 9 (Sep)
9	36 hours		33 hours	X	Month 10 (Oct)
A	1 day/24 hrs	M	Multiple hours	Y	Month 11 (Nov)
a	2 days/48 hrs	N	0 hrs, 1st guess	Z	Month 12 (Dec)

2.5 Deg Grid - Forecast Hours (h)

Code	Fcst Hrs/Days/	Code	Fcst Hrs/Days/	Code	Fcst Hrs/Days/
Value	Climo Mnth	Value	Climo Mnth	Value	Climo Mnth
0	*****		*****	0	
	*****	D	*****	P	
2	*****	E		Q	
3	*****	F		R	
4	*****	G		S	
5	*****	H			
6	*****	I	Initial	U	Update
	*****			V	
8	*****	K		W	
9	*****			X	
A	*****	M			
B	*****	N			

NOTES: ***** = These code values cannot be changed for 2.5 Deg Grid files.

8. i - Satellite Image File Contents Code

Code Value	GOES Files	DMSP Files	TIROS Files	SGDB Files
0	Vis 1:1	LS 1:1	Channel 1 1:1	N Vis 1:1
1	Vis 2:1	LS 2:1	Channel 1 2:1	(N Vis 2:1)
2	Vis 4:1	LS 4:1	Channel 1 4:1	(N Vis 4:1)
3	Vis 8:1		Channel 1 8:1	(N Vis 8:1)
4	Vis 16:1			
5	IR 1:1	TS 1:1	Channel 2 1:1	N IR 1:1
6	IR 2:1	TS 2:1	Channel 2 2:1	(N IR 2:1)
7	IR 4:1	TS 4:1	Channel 2 4:1	(N IR 4:1)
8			Channel 2 8:1	(N IR 8:1)
9				
A	MC 1:1	LF 1:1	Channel 3 1:1	S Vis 1:1
B	MC 2:1	LF 2:1	Channel 3 2:1	(S Vis 2:1)
C	MC 4:1	LF 4:1	Channel 3 4:1	(S Vis 4:1)
D		LF 8:1	Channel 3 8:1	(S Vis 8:1)
E		LF 16:1		
F	I	TF 1:1	Channel 4 1:1	S IR 1:1
G		TF 2:1	Channel 4 2:1	(S IR 2:1)
H		TF 4:1	Channel 4 4:1	(S IR 4:1)
I		TF 8:1	Channel 4 8:1	(S IR 8:1)
J	I	TF 16:1		
K			Channel 5 1:1	(T Vis 1:1)
L			Channel 5 2:1	(T Vis 2:1)
M			Channel 5 4:1	(T Vis 4:1)
N			Channel 5 5:1	(T Vis 8:1)
O				
P				T IR 1:1
Q				(T IR 2:1)
R				(T IR 4:1)
S				(T IR 8:1)
T				
U	Tropical Specialized Support (DMSP or TIROS)			

9. l - Atmospheric Level Code

Code Value	Level	Code Value	Level	Code Value	Level
0	1000 nb	D	50 nb	Q	
1	100 nb	E	30 nb	R	Gradient(1)
2	200 nb		20 nb	S	Surface
3	300 nb		10 nb	T	Tropopause
4	400 nb	H	HUD multi-lvl	U	Upper Lvl(2)
5	500 nb	I		V	
6	600 nb			W	
	700 nb	K	1000-500mb	X	
8	850 nb		1000-850mb	Y	SFC- 40K Ft
9	850-700mb	M	Multi-level	Z	40K-80K Ft
A	250 nb	N	850-500mb		
B	150 nb	O	SFC- 850nb		
	70 nb	P	700-500mb		

NOTES: (1) Gradient Level = SFC- 850nb
(2) Upper Level = 300-100mb

10. m - Grid Mesh Code

Code Value	Grid Mesh Projection	Code Value	Grid Mesh Projection	Code Value	Grid Mesh Projection
0	GADB	A		N	
1	NH Whole Mesh	B		O	
2	NH Half Mesh	C		P	
3	SH Whole Mesh	D		Q	
4	SH Half Mesh	E		R	
5	Trop Whole Mesh	F		S	
6	(2.5 Deg Grid) NH	G		T	(2.5 Deg Grid) LTL
7	(2.5 Deg Grid) SH	H		U	
8	NH Eighth Mesh	I		V	
9	SH Eighth Mesh			W	
T	(2.5 Deg Grid) LTL	K		X	
				Y	
		M		Z	

11. pp - Polar satellite identifier.

Code	Value	Description
	F8	DMSP UX9543
	F9	DMSP UX0542
	FA	DMSP (TBD)
	FB	DMSP (TBD)
	NA	NOAA NA0010
	NB	NOAA NA0011
	NC	NOAA NA0012
	ND	NOAA NA0013

12. q - Polar satellite quarter orbit identifier

Code	Value	Description
	A	Quarter orbit 1 (EQ->NP)
	B	Quarter orbit 2 (NP->EQ)
	C	Quarter orbit 3 (EQ->SP)
	D	Quarter orbit 4 (SP->EQ)

13. r - Polar satellite relative orbit identifier.

Code	Value	Description
	0	R+00
	1	R+01
	2	R+02
	3	R+03
	4	R+04
	5	R+05
	6	R+06
	7	R+07
	a	R+08
	9	R+09
	A	R+10
	B	R+11
	C	R+12
	D	R+13
	E	R+14

14. s - AFGWC/OTS Subsystem Code

Code		
Value	SSS	WF/OTS Subsystem
0	COT	Generic Subsystem (Chief, OTD)
1		(Open)
2		(Open)
3	US?	Special Projects Branch use
4		(Open)
5		(Open)
6		(Open)
7	csv	CONUS Severe
8		(Open)
9	HA2	Hazards Subsystem
A	***	APDS Test files and OTs (Harris)
B		(Open)
D	DEM	Data Base Managment
E	WSP	Snow Bogus
F	CHF	Chief Forecaster
G	css	Contingency Support/WFG
H	HWD	Horizontal Ueather
I	SAT	MetSat
J	JCS	Joint Chiefs of Staff support
K	***	Temporary use Satellite tests/Data Saves
L		(Open)
M	MRF	Medium Range
N		(Open)
O	WSP	RTNEPH Bogus
P	APD	AUDS Product Driver
Q	QCP	Quality Assurance
R	RSV	Mj Holliday (Reservist Studies)
S	SYN	Synoptician
T	TRO	Tropical
U	MDD	Mdels (Changed from Upper Air)
V		(Open)
W	WDF	WOFS/Low Level

X/Y/Z Reserved for use by contractor

15. t - Satellite Type Code

	CH 1	CH 2/VIS	CH 3	CH 4/IR	CH 5
NOAA	9 1	2	3	4	5
NOAA 10	6	7	8	8	0
NOAA 11	U	V	W	X	Y
F8	DMSP *	E	*	D	*
F9	DMSP *	B	*	C	*
GOES	*	H	*	G	*
SGDB	*	T	*	S	*

16. v - Vector File Content Code

Code Value	Surface Synoptic Analyses & Progs (1 = M,R,S,V)	Upper Air Analyses & Progs (1 = O-G,K,L,N,T,Y,Z)	HUD Multi-level Analyses & Progs (1 = H)
0			Red Ceiling Areas
1			Blue Ceiling Areas
2			Red Visibility Areas
3			Blue Visibility Areas
4	Pressure, 4mb intvl		Red Ceiling/Vsby Areas
5			Blue Ceiling/Vsby Areas
6			Areas-w/ CIGs 30-100
7			Areas w/ CIGs > 100
8	Pressure, 8mb intvl		
9			Trop. HWD/Sig Wx Input
A		Aircraft-derived Winds	
B			
C	Trop. Cyclones	Hgt Change, Falls	Cloud Free Areas
D		Divergence Axis	HUD Composite
E		Convergence Axis	
F		Thermal Trofs	Fog Areas
G			
H			High Cloud Areas
I	Isallobars		Icing Areas
J		Jetstream	
K			
L	Trop Cycln (Lat/Lon)	Streamlines	Low Level Wk (merged)
M	Trop Cycln (TMD)	Misture Axis	
N			
O	MWA Outlook Features		Obstructions to Vsby
P	Trop Special Support	Trop Special Support	Precipitation Areas
Q			
R	Trop Cycln (Recon,TMD)	RAOB-derived Winds	Heavy Rain Areas
S	Synoptic features (1)	Synoptic features (1)	Heavy Snow Areas
T	Trop. Syn. Fea. Input	Satellite-derived Winds	Thunderstorm Areas
U		Thermal Ridge Axis	Turbulence Areas
V		Thrmal Rdge & Cnvrge Axis	
U	Current MW features		Sig Wk Composite
X			
Y			
Z			Freezing Precip Areas

NOTES: (1) Synoptic features = fronts, pressure centers

17. y - Non-Standard Gridded Parameter Codes

0	Convergence	C	Convect Cond Level	0	Snow Depth
	Stretch Deformation	D	Lifted Cond Level	P	Potential Temp
2	Shear Deformation	E	Equilibrium Level	Q	Eq Potnl Temp
3	Vertical Shear	F	Level of Free Conv	R	Critical Temp
4	Total Deformation	G	Mbistr Divergence	S	Critical Heating
5		H	Thompson Index	T	Total Totals
6		I	Sfc Base Lifted Indx	U	Cross Totals
			Showalter Index	V	Vertical Totals
8		K	K-Index	W	Max Wind from RAOBS
9		L	Lifted Index	X	Hail Size
A	Avg Mxing Ratio	M	Mdified Lifted Indx	Y	
B	Wet Bulb Zero Hgt	N	Sweat Index	Z	

18. z - Product Vector File Content Code

Code Value	File Content	Code Value	File Content
	Icing SFC-10	A	Icing ABV 10
2	Turbc SFC-10	B	Turbc ABV 10
3	Tstns/Tps	C	SVR Weather (Non convective)
4	Cld Free	D	Areas of Cloud Depiction
5	Max Wnd (250mb)	E	Clds ABV 10
6	Thickness	F	SVR Weather (Convective)
7	Precipitation	G	SIG Precip (heavy)
8	Synoptic Features	H	SFC Press 8mb Isobaric Anal
9	M/FR (CIG/VSBY/Restrictions)	I	IFR (No Symbols)
0	IFR (CIG/VSBY/Restrictions)	K	Map Backgrounds
			Legend
		M	M/FR (No Symbols)
		S	Streamlines
		V	VSBY <5NM W/Restrictions

NOTES: Synoptic features = fronts, pressure centers, TC, Storm positions.

Operational Task Standards

O-S2, FILENAME CONVENTIONS, APPENDIX 3

Type of File	Type-ID	Logical Filename Contents and Examples
Animation File Lists	'ABL'	
AFOS	'AFOS'	map + title + valid-hr See FWDB Data Base Spec, Appendix 10 for standard AFOS logical filenames. EX: AFOS NA LMS POP-12 24H AFOS NA LFM 500 HGT 00H
Bogus	'BOGUS'	area + model/run + level + parameter + + nnH + format area = NA, NT, etc model/run = HIRAS, HIRASU, HIRAS7, HIRAS7U level = SFC, 250MB parameter = HGT, PRES nnH = Valid Hour format = TXT or VCT EX: BOGUS NT HIRAS 250 HGT 00H TXT BOGUS NT HIRASU SFC PRES 00H VCT
CATONE Plot Data	'CATONE'	form area level threshold CATONE plot and plot vectors EX: CATONE PLT NH SFC-40K CATONE VCT NH 40-80K GT 4
Computer Flight Plan	'CFP'	
Computer Flight Track	'CFT'	
Clipboard	'CLP'	abbreviated title See the Clipboard File Definition Doc EX: CLP EUR SFC ANALS
Programmable Cursor File	'CUR'	description

Directory Files	'DIR'	+ data base + comments FWDBA, FVDBB, GRID DB, PLOT DB, etc FWS RSDB
Function Key Definitions	'FKY'	+ operational task name + comments EX: FKY MFBELT
Generic Files Definitions	'GEN'	+ "ERIC" + comments EX: GENERIC IMAGE FOR IGM1
Gridded Data	'GRI'	+ mesh + area + data-item + level + + fcst-hr + (base-hr) mesh = UHL, HLF, EGT, GDE, HRS, 2.5 area = NH, SH, etc level = 1000MB, SURFC, 850MB, TROP, etc data-item = DVAL, MDVL, HGTS, TEMP, etc fcst-hr = 00H, 12H, 192H, etc (base-hr) = (-12), (-24), (-36), etc Default base-hr = (+00). See separate conventions for SGDB and Topography data file names. EX: GRI UHL NH DVAL 500MB 24H GRI HRS NH HGTS 1000MB 192H GRI GDB NT DVAL 500MB 12H (-12)
Gridded Data Plot Vectors	'GRPLOT'	+ remaining format as for Gridded Data EX: GRPLOT GDB NT TEMP 500MB
Stdndr Grid-Plot Vectors	'GRSTND'	+ remaining format as for Gridded Data EX: GRSTND GDB NT TEMP 500MB
Image	'IMG'	+ satt-type + area + image-content + comments + cycle satt-type = DMn D, GOE, COW TRn A GAC, SDB area = TRO, EUW USW etc. image-cant = TS 1:1, VS 16:1, C2 1:1, IR 2:1, etc comments = +MAPS, +TOPO, +TCn ANAL, etc cycle = (-06), (-72), etc A composite image is a combination of satellite imagery and merged overlay planes. EX: IMG DM2 D TRO TS 1:1 +MAPS +TC4 ANAL (-48)

Plot Data - Surface	'PLTSF'	+ area + parameters + projection + (base-hr) 1. Parameters required only if file has been edited or thresholded. 2. Base-hr of (+00) is optional. 3. Projection required only if plot file has been coordinate converted from lat/lon space to a specific map projection. EX: PLTSF EUT PRES WK SGD (-06) PLTSF EUT PRS/TMP/WINDS PST
Plot Data - SFC Vectors	'PLTSFV'	+ remaining format as for Surface Plot Data
Plot Data - Upper Air	'PLTUA'	+ area + level + parameters + projection + (base-hr) 1. Parameters required only if specific parameters have been edited. 2. Base-time of (+00) is optional 3. Level is required only if data for a single Level has been extracted with the SPP cmd. 4. Projection is required only if plot file has been coordinate converted from lat/lon space to a specific map projection. EX: PLTUA PA GOES WINDS TMD PLTUA UST 500MB WND
Plot Data - U/A Vectors	'PLTUAV'	+ remaining format as for Upper Air Plot Data EX: PLTUAV NH 300MB DPTD DATA
Print	'PRN'	+ comments Print files received from UNISYS systems.
Product Files (Text)	'PRT'	+ description NOT the same as RTOS messages (TTY products)
Product Files (Vector)	'PRV'	+ manop-heading + contents externally disseminated products.
Remapped Grid	'RMP'	+ polar satellite identifier + "R+ number" + quarter orbit EX: RMP F8 8 4
RTOS Message Form	'RIF'	+ msg-type + comments msg-type = ARP, ADM REQ, OTH RTOS Message Forms are templates for ARQs, REQs, TTY products, addressed messages, etc.

Image (Pseudo)	'IMS'	description A pseudo-image file is an image file created by merging overlay planes into refresh memory with out a satellite image. EX: IMS EUT SFC SYN FEA +L/L +GEO
Isopeteth Vectors (from grid data)	'ISO'	mesh t area + data-item + level + + fcst-hr + (base-hr) EX: ISO WHL NH DWAL 500MB 24H ISO HRS NH HGTS 300MB INTL ISO HVL NH THKN 85-50 00H (-18)
Library Data (SUOLIBE)	'LIB'	comments
List	'LST'	'PLT' or 'GRD' + logical-filename Plot or grid listings output from LST command. EX: LST GRI N WHL 50 DWL LST SFP USW
Lookup Table	'LUT'	image-type + curve + inclusions EX: LUT SGDB MB WMPS
Lookup Table (Enhancement Curve)	'LUT'	image-type t channel + curve + origin + season + purpose EX: LUT SGDB IR ED GVC ALL
Lookup Table (Pseudo-Image)/	'LUT'	PSI + purpose EX: LUT PSI NH PST MAP
Map Background	'MAP'	area + projection + resolution + mist-info See FWDB Data Base Spec, Appendix 20 EX: MAP N.HEM PST LNDSEA LORES MAP E.USA GOS GEOPOL HIRS
Operational Task Files	'OTF'	subsystem + title EX: OTF DEM Custom Mister Menu OTF SYN Bogus

RTOS Message	'RTS'	comments Includes ARQs, REQs, addressed messages
Plot Model Templates	'TEM'	type + description type = ACFT, CATONE, SFC, TROP, UA, etc. See FWDB Data Base Spec, Appendix 23
Temporary File	'TMP'	logical-filename Code the Logical filename according to the format for the file type and prefix 'TMP'
Toggle Data	'TOG'	polar satellite identifier + "R+ number" + quarter orbit + data type EX: TOG F8 8 4 CTA
Topography	'TOPO'	area + projection + format + misc format = 'ISO' or 'IMG' EX: TOPO USW PST ISO TOPO USW SGD IMG
Toggle Template	'TTM'	data type EX: TTM DEP
Teletype Product (AFGWC)	'TTY'	nanop-heading + contents +...+ Manopd + OPR 1. TTY Products are RTOS files. 2. Omit 'KGWC' from nanop-heading in Logical filename. EX: TTY FAEU EUR LOLVL FLIGHT HAZARDS
Text Files	'TXT'	subsystemid + description EX: TXT SYN SYNOPTICIAN QC REPORT
UNISYS Files	'UNI'	description
Vector	'VCT'	area + level + contents + fcst-hr area = NH, USW, AST, TAP, etc. level = SFC, 1000, 850, TROP, etc. contents = parameters or description fcst-hrs = 06H, 12H, 24H, etc. 1. Internal vector files, NOT vector product source files. EX: VCT USW SFC HVY PRECIP AREAS

Logical File Type Identifiers

TypeID - Description

ABL - Animation File Lists
AFOS * AFOS Products
BOGUS - Bogus Files
CATONE - Catone Reports
CFP - Computer Flight Plans
CLP * Clipboard Files
CUR - Cursor Definition Files
DIR - Directory Files
FKY - Function Key Definition Files
GENERIC- Generic Files
GRI - Gridded Data Files
GRPLOT - Gridded Plot Files
GRSTND * Standard Gridded Data Files
IMG - Image Files
IMS - Pseudo Image Files
ISO - Isopleth Files
LIB - Library Files (SUOLIBE)
LST - List Output Files
LUT * Enhancement (Lookup) Tables
MAP - Map Backgrounds
OTF - Operational Task Files
PLT - Plot / Plot Vector Files
PRF - Teletype Product Files
PRN - Print Files
PRI - Text Product Files (NOT RTOS)
PRV * Vector Product Files
RMP - Remapped Grid
RTOS - RTOS Message Forms
RTS - RTOS Messages
TEM - Plot Model Templates
TMP - Temporary Files
TOG - Toggle Data
TOPO - Topography Files
TIM - Toggle Template
TTY - Teletype Product Files (Bltns)
TXT - Text Files (Forms/Stat/Dynamic)
UNI - UNIVAC Batch Jobs
VCT - Vector Files

* These File Types require DBA coordination prior to use.

FROM HQ AFGWC/SYS/Library

20 February 1992

SUBJECT: Software Standard Change Proposal M-S1

TO: SDS
SSWG
IN TURN

Proposal:

1. Paragraph 3.c., first line, DELETE the word "tape".
2. Paragraph DELETE the entire paragraph.

Why do we need this?:

1. This standard deals with file naming conventions for disk resident subsystem files, not tape files.
2. A means of system control (i.e., residing on tape) should be covered in a subsequent Software Configuration Management Standard AND should have the concurrence of all subsystem managers. After all, it's for control of their subsystems.

What are the benefits?:

Elimination of ambiguity and overlap of Software Configuration Management Standards.